SelfServe New Stack Business Services

Interface Design Document

Fred Caprilli

Erick Hu

Kevin Lo

Ivaylo Gabrovsky

March 13, 2017

Version: 1.102

Contents

[Introduction 6](#_Toc467760690)

[SelfServe "New Stack" 6](#_Toc467760691)

[Architecture 6](#_Toc467760692)

[Error handling 7](#_Toc467760693)

[Versioning 7](#_Toc467760694)

[Business Domains 7](#_Toc467760695)

[Interface Specification Approach 7](#_Toc467760696)

[REST Request Format 7](#_Toc467760697)

[JSON Response Format 8](#_Toc467760698)

[UsageManagement 3.0, 3.1, 3.2, 3.3 REST Service 9](#_Toc467760699)

[bill-cycle-info 9](#_Toc467760700)

[bill-cycles 11](#_Toc467760701)

[data-and-messaging-meters 12](#_Toc467760702)

[voice-meters 16](#_Toc467760703)

[internet-meters-and-details 19](#_Toc467760704)

[service-types 22](#_Toc467760705)

[summary 24](#_Toc467760706)

[detail 26](#_Toc467760707)

[usage 28](#_Toc467760708)

[transactions 30](#_Toc467760709)

[refunds 33](#_Toc467760710)

[partners 35](#_Toc467760711)

[top-usage 37](#_Toc467760712)

[prepaid-meters 39](#_Toc467760713)

[UsageMeter REST Service 41](#_Toc467760714)

[detailed-data-and-messaging-meters 41](#_Toc467760715)

[shared-data-and-messaging-usage 49](#_Toc467760716)

[UsageMeter 2.0 REST Service 52](#_Toc467760717)

[detailed-data-and-messaging-meters 52](#_Toc467760718)

[shared-data-and-messaging-usage 59](#_Toc467760719)

[UsageDetail 1.0/2.0 REST Service 61](#_Toc467760720)

[voice-detail 61](#_Toc467760721)

[cached-call-detail 65](#_Toc467760722)

[UsagePreferenceManagement REST Service 66](#_Toc467760723)

[subscriber-threshold-preferences 66](#_Toc467760724)

[subscriber-threshold-preferences (PUT) 73](#_Toc467760725)

[subscriber-notification-preferences 79](#_Toc467760726)

[subscriber-notification-preferences (PUT) 82](#_Toc467760727)

[subscriber-payperuse-preferences (PUT) 85](#_Toc467760728)

[validate-sms-number 88](#_Toc467760729)

[UsagePreferenceManagement 1.1 REST Service 89](#_Toc467760730)

[subscriber-threshold-preferences 89](#_Toc467760731)

[UsageBlockManagement REST Service 96](#_Toc467760732)

[data-block-status 96](#_Toc467760733)

[data-block-status (PUT) 99](#_Toc467760734)

[UsageNotificationAcceptanceHistory REST Service 102](#_Toc467760735)

[threshold-history 102](#_Toc467760736)

[notification-contact-change-history 107](#_Toc467760737)

[notification-preference-history 108](#_Toc467760738)

[notification-history 111](#_Toc467760739)

[block-consent-history 114](#_Toc467760740)

[block-event-history 116](#_Toc467760741)

[PaymentManagement 3.0 REST Service 118](#_Toc467760742)

[payment-info 118](#_Toc467760743)

[payment-method 121](#_Toc467760744)

[PaymentProcessing REST Service 123](#_Toc467760745)

[payment-with-registered-card 123](#_Toc467760746)

[payment-with-unregistered-card 126](#_Toc467760747)

[card-authorization 128](#_Toc467760748)

[PrepaidPaymentManagement REST Service 131](#_Toc467760749)

[prepaid-info 131](#_Toc467760750)

[topup-settings 133](#_Toc467760751)

[prepaid-balance (POST) 134](#_Toc467760752)

[prepaid-auto-topup (PUT) 136](#_Toc467760753)

[payment-card (PUT) 138](#_Toc467760754)

[AccountManagement REST Service 140](#_Toc467760755)

[account-pin 140](#_Toc467760756)

[billing-address (GET) 142](#_Toc467760757)

[valid-address 144](#_Toc467760758)

[billing-address (PUT) 146](#_Toc467760759)

[AccountActivity REST Service 148](#_Toc467760760)

[account/transaction-history 148](#_Toc467760761)

[account/payments 151](#_Toc467760762)

[sub/transaction-history 153](#_Toc467760763)

[PrepaidActivity REST Service 156](#_Toc467760764)

[prepaid-call-and-event-history 156](#_Toc467760765)

[cached-prepaid-call-and-event-history 162](#_Toc467760766)

[AccountInfo 2.1/2.2/2.3 REST Service 163](#_Toc467760767)

[account (GET) 163](#_Toc467760768)

[account-summaries 166](#_Toc467760769)

[ebill-subscription (GET) 177](#_Toc467760770)

[self-serve-subscription (GET) 178](#_Toc467760771)

[credit-info (GET) 179](#_Toc467760772)

[CustomerInfo REST Service 180](#_Toc467760773)

[summaries 180](#_Toc467760774)

[search (POST) 189](#_Toc467760775)

[search (GET) 192](#_Toc467760776)

[business-hierarchy (Obsolete) 194](#_Toc467760777)

[business-account-summaries 196](#_Toc467760778)

[business-account-hierarchy 199](#_Toc467760779)

[account-status 201](#_Toc467760780)

[CreditInfo REST Service 203](#_Toc467760781)

[details 203](#_Toc467760782)

[SubscriberInfo REST Service 204](#_Toc467760783)

[subscriber (GET) 204](#_Toc467760784)

[self-serve-subscription (GET) 206](#_Toc467760785)

[payment-responsible-status (GET) 208](#_Toc467760786)

[InvoiceInfo REST Service 209](#_Toc467760787)

[bill-summary 209](#_Toc467760788)

[service-summary 218](#_Toc467760789)

[subscriber-summary 225](#_Toc467760790)

[bill-archives 229](#_Toc467760791)

[bill-archive-url 231](#_Toc467760792)

[video-bill 232](#_Toc467760793)

[bill-delivery-preference 233](#_Toc467760794)

[update-bill-delivery-preference 234](#_Toc467760795)

[group-summary 235](#_Toc467760796)

[restricted-tax-credit-report 238](#_Toc467760797)

[Appendix 241](#_Toc467760798)

[Error Codes 241](#_Toc467760799)

[Decommissioned Services 242](#_Toc467760800)

# Introduction

## SelfServe "New Stack"

The current SelfServe architecture consists of a monolithic Struts web app encompassing over 30 workflows. Even small changes require a rebuild, retest, redeploy cycle. Over the years the code has become increasingly difficult to maintain, due to support for obsolete functionality which has never been removed, as well as less than ideal separation of presentation from business logic.

In order to align with new priorities including the ability to make and deploy changes quickly, the existing application 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), currently referred to as *beta.telus.com*. We refer to this suite of services as "SelfServe New Stack" (SSNS).

This document is an interface specification of these services.

## Architecture

SSNS services are conceived as business-level services. That is, they provide functionality at a level suitable for a UI web app to consume directly. Such services typically aggregate information from and orchestrate functionality from lower level domain web services. As such, 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. Where possible, we will leverage the new "Data Grid" – our new caching layer – to obtain common account- and subscriber-level information, in order to maximize performance. In some of the more complex flows, such as price plan and services management, we cache complex objects to avoid passing large structures across multiple layers, which would otherwise be necessary to maintain state in a purely stateless world.

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 Integrated Architecture (formerly known as 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. 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.

## 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 | IB (invalid ban), IS (invalid sub)… |
| 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.

## Business Domains

The SelfServe list of functions has been grouped into five broad domains:

* Usage Management
* Payment Management
* Account Management
* Device Management
* Plans & Services Management

Each service will be named based on one of these domains. There may be more than one service per domain.

## 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.

## REST Request Format

The guiding principle here is to keep the URL simple and unambiguous. Hence:

**Operations and parameter names are all lowercase.** This avoids ambiguities introduced by mixed case. Which leads to:

**Operation resource names that are compound words can be hyphenated** e.g. *price-plan-eligibility*.

**Compound words do not use camel caps**. *e.*g. *billcycleinfo*, not *billCycleInfo*.

**Boolean-style flags are y/n.** e.g*., totalsonly=y*, not *totalsonly=true*.

The above relates to the URI itself only. **Parameters within a JSON structure in the request payload (i.e. for methods other than GET) should follow the camel case convention.**

Note that this is for input only. Response format is discussed in the next section.

Example:

*{domain}/usagemanagment/account/342554/sub/4165556721/data-and-messaging-meters/billcycle=39&decorate=y*

In the above example, data-and-messaging-meters is the operation. Note – all lowercase, no camel caps. Operation name is hyphenated, parameters, e.g. billcycle, are not. Boolean decorate is y, not “true”.

## JSON Response Format

The JSON response returned will follow some common standards:

**Elements will null values will not be included in the response**. JSON style guides such as Google's favour this approach as it keeps the payload as concise as possible. This may be relaxed where it makes sense to do so.

**Compound words use camel caps.** E.g. "firstName"

**String values are enclosed in double quotes**. E.g. "firstName": "*John*", "localizedAmt": "*37,25 $*"

**Numeric values are unquoted**. E.g. "dollarAmt": *37.25*

**Boolean values are true/false**. E.g. "recurringInd" : *true*

**Date/time values are in ISO 8601 format**. E.g. "eventDt": "*2013-11-03T13:07:05Z*"

**Date elements which don't have a time component are in yyyy-mm-dd format**. E.g. "billCycleStartDt": "*2012-07-05*".

# UsageManagement 3.0, 3.1, 3.2, 3.3 REST Service

This service provides operations in support of displaying summary and detailed usage information, including a "usage dashboard" with graphical meters representing usage against an allowance.

## bill-cycle-info

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-cycle-info  account/{ban}/sub/{sub}/bill-cycle-info | |
| **Method** | GET |
| **Description** | Basic bill cycle info for usage dashboard |
| **Input** | **ban** – account number (currently only wireless supported; postpaid only}  **sub** – phone number – this is the newer interface and is the preferred one. Even though the sub is not technically required here, this operation is related to obtaining usage at the subscriber level  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated.  **NOTE: decoration not currently implemented** | | language | "en", "fr" | Optional | Language of decoration. Default is "en". | | appid | Indicates which set of resources to decorate with | Optional | Identifies apps with potentially different sets of verbiage; this verbiage would be maintained in RefPDS; e.g. "web-telus", "device-telus", "web-koodo", "device-koodo" |   Examples:  UsageManagementREST/account/2343465/bill-cycle-info  UsageManagementREST/account/2343465/bill-cycle-info?decorate=y&language=en&appid=koodo-web |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN | BAN [90653242] Not Found in Datagrid |  |  | | 400 | IBT | Invalid BAN type (prepaid) |  |  |  | | 500 |  | AccountInformationService call failed | underlying error code | underlying error message | Any other Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billCycleInfoResponse" : {  "billCycleInfo": {<billCycleInfo>},  "status” : <status>  Where <billCycleInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | endDt | date | end date of cycle | e.g. “2013-06-07T08:50:44” |  | | startDt | date | start date of cycle | e.g. “2013-05-08T08:50:44" |  | | endDateTxt | String | Localized date | e.g. “23 Juin, 2013” | x | | startDateTxt | String | Localized date | e.g. “June 23, 2013” | x | | daysElapsed | integer | number of days into the cycle | 17 |  | | daysInCycle | integer | number of days in cycle | e.g. 30, 31 |  | | billCycleCd | Integer | Bill cycle number | 22 |  | | captionTxt | string | Caption for cycle info | “Bill cycle”, “Cycle de facturation” | x | | endsTxt | String | Template for “ends” wording | “ends {0}“, “finit le {0}“ | x |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | "billCycleInfoResponse": {  "billCycleInfo" :  { "endDt":"2013-06-09T08:50:44",  "startDt":"2013-05-10T08:50:44",  "endDateTxt": null,  "startDateTxt": null,  "daysElapsed":27,  "daysInCycle":31,  "billCycleCd":22,  “captionTxt” : null,  “endsTxt” : null  },  status :  { "statusCd" : "200",  "statusTxt" : "OK",  } | billCycleInfo :  { "endDt":"2013-06-09T08:50:44",  "startDt":"2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 10, 2013",  "daysElapsed":27,  "daysInCycle":31,  "billCycleCd":22,  “captionTxt” : "Bill cycle",  “endsTxt” : "Ends {0}"  },  Status :  { “statusCd” : 200,  “statusTxt” : “OK”,  } | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.2**  getAccountByAccountNumber | 700 ms average; 3000 ms peak |   Expected average response time: 1000 ms |
| **Sample GUI** |  |

## bill-cycles

|  |  |
| --- | --- |
| **OPERATION**  bill-cycles | |
| **Method** | GET |
| **Description** | list of bill cycle start/end dates |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Mandatory | Billing system | | cyclecode | number | Mandatory |  | | cycles | 1-12 | Optional | Default is 3 | |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid system |  |  | must be "wls" or "ffh" | | 400 |  | Invalid cyclecode |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billCyclesResponse" : {  "billCycleList": {<billCycle>},  "status” : <status>  Where <billCycle> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | cycleNum | integer | ordinal number of cycle | 0, -1, -2, -3… |  | | endDt | date | end date of cycle | e.g. “2013-06-07” |  | | startDt | date | start date of cycle | e.g. “2013-05-08" |  |   Example:  "billCyclesResponse": {  "billCycleList" : [  { "cycleNum : 0 ",  "endDt":"2014-04-09",  "startDt":"2014-03-10"  },  { "cycleNum : -1 ",  "endDt":"2013-03-09 ",  "startDt":"2013-02-10"  },  { "cycleNum : -2 ",  "endDt":"2013-02-09",  "startDt":"2013-01-10"  },  status :  { "statusCd" : "200",  "statusTxt" : "OK",  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | |  |  |   Expected average response time: 500 ms |
| **Sample GUI** |  |

## data-and-messaging-meters

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/data-and-messaging-meters | |
| **Method** | GET |
| **Description** | Data and Messaging meters for current bill cycle |
| **Input** | **ban** – account number (currently only wireless supported; postpaid only}  **sub** – subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | subscriberid | Subscriber ID | Optional | When provided, implementation can bypass call to SubscriberInfoService. | | billcycle | Bill cycle number | Optional | When these are both provided, implementation can bypass call to AccountInformationService. It can be obtained by the initial call to **bill-cycle-info**. | | enddate | Bill cycle end date  yyyy-mm-dd | | cycleperiod | which bill cycle to return – current, previous, two back | Optional | Possible values are **0** (default), **-1** (previous bill cycle), **-2** (2 cycles back) | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="n"), response will not be decorated. **(PARTIALLY implemented – see below)** | | language | "en", "fr" | Optional | Language of decoration. Default is "en". If decorate parameter is missing, this will be ignored. | | appid | Indicates which set of resources to decorate with | Optional | "web-telus", "device-telus", "web-koodo", "device-koodo" **(NOT implemented)** | | passportexpiry | Indicates whether to provide expiry date for BCIC socs | Optional | will require an extra downstream call to get the expiry date in some cases | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN (AIS call failed) | underlying error code | underlying error message | AIS fails due to invalid BAN – AIS errors:  BAN [90653242] Not Found  messageId=APP10004, errorCode=CMB\_AIS\_0001 | | 400 | IS | Invalid Sub (SIS call failed) | underlying error code | underlying error message | SIS fails due to invalid sub – SIS errors: Subscriber not found for phone number: 4165765688;  messageId= APP20002,  errorCode=CMB\_SIS\_0015 | | 400 | ISI | Invalid Sub ID (CRI call failed) | underlying error code | underlying error message | CRI invalid sub error:  CRI-002002: "Invalid sub"; messageId= CRI-002002,  errorCode=CRI-002002 | | 400 | IBC | Invalid bill cycle (CRI call failed) | underlying error code | underlying error message | CRI invalid bill cycle. Include error msg which indicates correct bill cycle: CRI-000016: "Invalid bill cycle"; messageId= CRI-000016,  errorCode= CRI-000016,  errorMessage=xxxx | | 500 |  | CustomerRatingInquiryService call failed | underlying error code | underlying error message | Any other CRI exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | “dataAndMessagingMetersResponse" :  {  "data": {  "meterInfo" : <meter info>}  },  "messaging": {  "meterInfo" : <meter info>}  }  },  "status" : <status>  where <meter info> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterList | array of <meter> | see below |  |  | | totalChargedAmt | decimal | dollar amount | 3.45, 17.00 |  | | totalChargedTxt | string | Localized dollar amount | "$3.45", "17,00 $" | x | | totalChargedDescTxt | string | "charged" description | "Total charged amount: " | x | | lastUpdatedDt | date | ISO8601 format date/time | "2013-05-22T05:07:22Z" |  | | lastUpdatedTZ | string | time zone | "PST", "EST" |  | | lastUpdatedTxt | string | Localized date corrected for timezone and daylight savings time | "Jun 6, 2013 8:37:04 EDT",  "6 Juin 2013 8:37:04 HNA", | x | | lastUpdatedDescTxt | string | description template | "Last updated: {0}", "Dernière mise à jour le {0}" | x | | sectionHeadingTxt | string | Data section heading | "Data", "Données", "Messaging" | x |   and <meter> is a structure with the following elements:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | typeCd | string | meter type | "allowance", "unlimited", "ppu", "tier", "share", "datasaver" |  | | meterNum | integer | ordinal | e.g. 1, 2, 3… |  | | serviceCd | string | soc that meter is attached to | e.g. "SXCC10C", etc. |  | | serviceDescTxt | string | soc description | "2GB Data" | x | | groupCd | string | area of usage | "C" (Canada), "CU" (Canada-US), "U" (US), "I" (International), "A" (All) |  | | regionCd | string | domestic vs. roaming | "N" (Nationwide), "R" (Roaming), "A" (All) |  | | usedAmt | double | consumed amount | e.g. 512.0 (MB), 23.0 (messages) |  | | includedAmt | double | allowance | e.g. 2048.0 MB, 100.0 messages |  | | chargedAmt | double | Dollar amount charged | e.g. 2.40 |  | | unitOfMeasureCd | string | units | "MB", "msgs" |  | | unitOfMeasureTxt | string | Localized unit | e.g. "MB", "Mo", "Messages" | x | | directionCd | string | direction of SMS | "S" (sent), "R" (received), "SR" (sent/received), empty (n/a – e.g. for data) |  | | tierNum | integer | tier number (for tiered meters only) | e.g. 0, 1… |  | | statusCd | string | active vs. expired | "active", "expired" |  | | timestampTxt | string | ISO8601 format date/time | "2013-05-22T05:07:22Z" |  | | bcicInd | boolean | indicates whether soc is a BCIC soc (Roaming passport) | true, false |  | | captionTxt | string | Caption | "Nationwide", "US", "International" | x | | subCaptionTxt | string | Optional text to append to caption | "(current)", "(previous)" | x | | modeCd | string | Graph mode representing business rule for when to present as bar graph | "graph", "no-graph", "no-usage", "unavailable" | x | | blurbTxt | string | Text to display below the meter | "Usage above only represents your current usage as part of your share plan and does not reflect accumulated usage from other share plan members. " | x | | effectiveDt | String | effective date of soc for this meter |  |  | | expiryDt | String | expiry date of soc for this meter |  |  | | aggregateInd | boolean | indicates whether this meter is an aggregate of 2 or more raw meters. Used for MMS. |  |  | | pricePlanInd | boolean | indicates whether the soc for this meter is a price plan (as opposed to an add-on) |  |  |   Items in red are not currently implemented.  Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | "dataAndMessagingMeters" : {  "data": {  "meterList": [  {  "typeCd": "allowance",  "meterNum": 1,  "serviceCd": "SXCC10C",  "groupCd": "C",  "regionCd", "N",  "usedAmt": 864,  "includedAmt": 2048,  "chargedAmt" : 0.0,  "unitOfMeasureCd": "MB",  "amount": 0.0,  "directionCd": null,  "tierNum": 0,  "statusCd": "active",  "timestampTxt": "2013-05-22T05:07:23Z",  "bcicInd": false  }  ]  },  "totalChargedAmt": 1.40,  "lastUpdatedDt": "2013-05-22T05:07:23Z",  "lastUpdatedTZ": "EST",  },  "messaging": {  "meterList": [  {…  },  "status" : {… } | "dataAndMessagingMeters" : {  "data": {  "meterList": [  {  "typeCd": "allowance",  "meterNum": 1,  "serviceCd": "SXCC10C",  "serviceDescTxt": "Favourite Numbers 10",  "groupCd": "C",  "regionCd", "N",  "usedAmt": 864,  "includedAmt": 2048,  "chargedAmt" : 0.0,  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "Mo",  "amount": 0.0,  "directionCd": null,  "tierNum": 0,  "statusCd": "active",  "timestampTxt": "2013-05-22T05:07:23Z",  "bcicInd": false,  "captionTxt": "États-Unis",  "subCaptionTxt": "(actuel)",  "modeCd": "graph",  "blurbTxt": "Votre forfait inclut 2048 Mo"  }  ]  },  "totalChargedAmt": 1.40,  "totalChargedTxt": "1,40 $",  "totalChargedDescTxt": "Montant chargé:",  "lastUpdatedDt": "2013-05-22T05:07:23Z",  "lastUpdatedTZ": "EST",  "lastUpdatedTxt":"6 Juin 2013 8:38:07 HNA",  "lastUpdatedDescTxt": "Mise à jour:",  "sectionHeadingTxt": "Données"  },  "messaging": {  "meterList": [  {…  },  "status" : {… } | |
| **SLA** | |  |  | | --- | --- | | **Service/operation** | **SLA** | | **AccountInformationService 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak | | **SubscriberInformationService 3.0**  getSubscriberIdentifierByPhoneNumberAndAccountNumber | 2000 ms average; 5000 ms peak | | **CustomerRatingInquiryService 3.0**  getMeterList | 1500 ms average; 3000 ms peak | | **ServiceOrderReferenceService v 4.3** getRegularService  getPricePlan | 1000 ms average |   Expected average response time: 1500 – 5000 ms (depending on which optional parameters are passed in).  Note: SLAs for services above are based on numbers entered into SOA Repository Manager, and represent an average. In most cases we're calling the more lightweight operations on those services, which we anticipate will have more favourable response times. Hence, the estimated response for this service is probably a worst case scenario. |
| **Sample GUI** |  |

## voice-meters

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/voice-meters | |
| **Method** | GET |
| **Description** | Airtime usage information for current bill cycle (unbilled) |
| **Input** | **ban** – account number (postpaid only)  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated. **NOT CURRENTLY IMPLEMENTED.** | | language | "en", "fr" | Optional | Language of decoration. Default is "en". | | appid | Indicates which set of resources to decorate with | Optional | Indicates which verbiage set to use: "web-telus", "device-telus", "web-koodo", "device-koodo" |   Examples:  UsageManagementREST/account/2343465/sub/4165765687/voice-meters  UsageManagementREST/account/2343465/sub/4165765687/voice-meters?decorate=Y&language=fr&appid=web-telus |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN or sub; or no data available | BAN [90653242] Not Found | messageId=SYS00003, errorCode=  CMB\_WRAUIS\_0001 | WRAUI throws Service Exception not Policy Exception; bad BAN or sub (or missing data) | | 500 |  | Wireless Rated Airtime Usage Inquiry Svc failed | underlying error code | underlying error message | Any other Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “voiceMetersResponse” : {  "meterList" : [<voicemeter>],  <meter info>  },  "status” : <status>  where <voicemeter> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | serviceCd | string | soc that meter is attached to | “SXCC10C”, etc. |  | | unitOfMeasureCd | string | Minutes or seconds (currently only minutes) | "M", "S" |  | | unitOfMeasureTxt | string | localized units | "minutes", "seconds", "secondes" | x | | usedAmt | int | amount used from allowance | 34 |  | | totalUsedAmt | int | total amount used (including free) | 35 |  | | includedAmt | int | allowance | 200 |  | | freeUsedAmt | int | free minutes used | 20 |  | | chargedAmt | int | overage minutes | 15 |  | | chargedDollarAmt | double | overage fees | 12.75, 5.00 |  | | chargedDollarTxt | string | localized dollar amount | "$12.75", "5,00 $" | x | | unlimitedInd | boolean | indicates whether is unlimited | “Y”, “N” |  | | statusCd | string | active vs. expired | "active", "expired" |  | | timestampTxt | string | ISO8601 format date/time | "2013-05-22T05:07:22Z" |  | | captionTxt | string | Caption | "Nationwide" | x | | subCaptionTxt | string | Optional text to append to caption | "(current)", "(previous)" | x | | modeCd | string | Graph mode representing business rule for when to present as bar graph | "graph", "no-graph", "no-usage", "unavailable" | x | | blurbTxt | string | Text to display below the meter | "Usage associated with certain add-on minutes may not be reflected in these numbers." | x |   and where <meter info> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | totalChargedAmt | decimal | dollar amount | 3.45, 17.00 |  | | totalChargedTxt | string | Localized dollar amount | "$3.45", "17,00 $" | x | | totalChargedDescTxt | string | "charged" description | "Total charged amount: " | x | | lastUpdatedDt | date | ISO8601 format date/time | "2013-05-22T05:07:22Z" |  | | lastUpdatedTZ | string | time zone | "PST", "EST" |  | | lastUpdatedTxt | string | Localized date corrected for timezone and daylight savings time | "Jun 6, 2013 8:37:04 EDT",  "6 Juin 2013 8:37:04 HNA", | x | | lastUpdatedDescTxt | string | description template | "Last updated: {0}", "Dernière mise à jour le {0}" | x | | sectionHeadingTxt | string | Data section heading | "Data", "Données", "Messaging" | x |   NOTE: Currently we're unable to obtain information about how recently the usage figures were updated for voice. (The UI will typically display a generic message indicating "within the last 24 hours"). These fields are replicated here from data-and-messaging-meters to allow for the possibility of such information being available in the future.  Example:   |  |  | | --- | --- | | Undecorated | Decorated | | {  "voiceMetersResponse": {  "voice": {  "meterList": {  "serviceCd": "PTKNA1000",  "unitOfMeasureCd": "M",  "usedAmt": 26,  "totalUsedAmt": 29,  "includedAmt": 50,  "freeUsedAmt": 3,  "chargedAmt": 15,  "chargedDollarAmt": 3.00,  "unlimitedInd": "N",  "statusCd": "active",  "timestampTxt": "2013-05-22T05:07:23Z",  "modeCd": "graph"  },  "totalChargedAmt": 1.40,  "lastUpdatedDt": "2013-05-22T05:07:23Z",  "lastUpdatedTZ": "EST"  },  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }  } | {  "voiceMetersResponse": {  "voice": {  "meterList": {  "serviceCd": "PTKNA1000",  "unitOfMeasureCd": "M",  "unitOfMeasureTxt": "",  "usedAmt": 26,  "totalUsedAmt": 29,  "includedAmt": 50,  "freeUsedAmt": 3,  "chargedAmt": 15,  "chargedDollarAmt": 3.00,  "chargedDollarTxt": "$3.00",  "unlimitedInd": "N",  "statusCd": "active",  "timestampTxt": "2013-05-22T05:07:23Z",  "captionTxt": "Local",  "subCaptionTxt": null,  "modeCd": "graph",  "blurbTxt": "Your plan includes 50 min."  },  "totalChargedAmt": 1.40,  "totalChargedTxt": null,  "totalChargedDescTxt": null,  "lastUpdatedDt": "2013-05-22T05:07:23Z",  "lastUpdatedTZ": "EST",  "lastUpdatedTxt": null,  "lastUpdatedDescTxt": null,  "sectionHeadingTxt": "Airtime minutes"  },  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }  } | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessRatedAirtimeUsageInquiryService v 1.0**  getAirtimeUsageSummary | 2000 ms average; 5000 ms peak |   Expected average response time: 2000 ms |
| **Sample GUI** |  |

## internet-meters-and-details

Changes for version 3.3 highlighted in yellow.

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/instance/{instanceid}/internet-meters-and-details | |
| **Method** | GET |
| **Description** | HSIA usage info |
| **Input** | **ban** – FFH account number  **instanceid** – ID of the specific service instance. Use the **instanceId** from the Product returned by CustomerInfo/summaries;  Parameters:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | productbillingid | identifies the product instance | Mandatory | Alternative identifier of the product instance (like instanceid in the URI); required for current cycle usage; this is returned alongside instanceId by CustomerInfo/summaries. | | cycleperiod**s** | comma-delimited list of bill cycles for which usage is required | Optional | Possible values are **0** (default), **-1** (previous bill cycle), **-2** (2 cycles back), etc. up to **-13**  e.g. -2,-4,-6. | | cyclerange | start cycle and end cycle separated by comma | Optional | e.g. -1,-3 – will return the three cycles starting with previous bill cycle | | billcycle | Bill cycle number | Mandatory | bill cycle code is returned by **account-summaries** | | planmode | type of plan description – default or Bill Presentment | Optional | Default is default (**DC**); specify **BP** for Bill Presentment description | | details | include detail rows | Optional | y/n – set to n to exclude daily details | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="n"), response will not be decorated. | |  |  |  |  | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 200 | NODATA | No data found |  |  | BAN valid but no match on instanceId; empty result set from back end | | 400 |  | Billcycle must be between 1 and 28  Must enter either cycleperiods or cyclerange  Must enter either cycleperiods or cyclerange. NOT both  Cycleperiods must between -13 to 0  Cyclerange must between -13 to 0  Cyclerange must be in this format (startcycleperiod, endcycleperiod) |  |  | invalid cycleperiod, bill cycle | | 400 |  | ProductBillingId is required |  |  | Product Billing id is required. | | 400 | IB |  | underlying error code | underlying error message | BAN not found | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | “internetMetersAndDetailsResponse" :  {  "instanceId" : <integer>,  "billCycleUsageList" : [<billCycleUsage>],  "status" : <status>  }  where < billCycleUsage> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | startDt | Date | start date of cycle |  |  | | endDt | Date | end date of cycle |  |  | | lastUpdatedDt | timestamp | Date/time of last update; only relevant for current bill cycle | populated for current bill cycle only |  | | meterInfoList | [meterInfo] | list of meterInfo | Normally there will only be one element in this list; however if there has been a price plan change during the cycle there would be two meters, one representing the date range within the cycle corresponding to each plan. |  |   where <meterInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | startDt | Date |  |  |  | | endDt | Date |  |  |  | | typeCd | string | type of meter | "allowance", "unlimited" |  | | downloadUsedAmt | double |  |  |  | | uploadUsedAmt | double |  |  |  | | totalUsedAmt | double |  |  |  | | unitOfMeasureCd | String |  | "GB" |  | | totalChargedAmt | double | dollar amount charged |  |  | | planInfo | <planInfo> |  |  |  | | usageDetailList | [<usagedetail>] |  |  |  |   where <planInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | planTxt | string | plan description |  |  | | planIncludedAmt | double | allowance | 25 |  | | unitOfMeasureCd | string | unit of measure | GB |  | | addOnList | [addOn] | list of add-ons |  |  |   where <addon> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | addOnTxt | string | add-on description |  |  | | addOnAmt | double | allowance | 25 (will be 0 if unlimited) |  | | unlimitedInd | boolean | indicates unlimited usage | GB |  |   where < usage detail> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | usageDt | date | date of usage | "2014-01-24" |  | | uploadAmt | double | amount uploaded in KB | 4000 (i.e. 4 MB) |  | | downloadAmt | double | amount downloaded in KB | 2000 |  | | totalAmt | double | sum of upload and download in KB | 6000 |  | | netChargeAmt | double | dollar amount of overage | 4.3 |  | | pstAmt | double | PST charged on overage | 0.34 |  | | gstAmt | double | GST charged on overage | 0.47 |  | | hstAmt | double | HST charged on overage | 0.81 |  | | discountAmt | double | discount applied | 0.25 |  | | totalChargeAmt | double | net plus taxes minus discount | 5.11 |  | | billedInd | boolean | indicates whether was billed | true, false |  |   Example:  {  "internetMetersAndDetailsResponse": {  "status": {  "ns4:statusCd": 200,  "ns4:statusTxt": "OK"  },  "instanceid": 602223804,  "billCycleUsageList": [  {  "startDt": "2014-10-28T00:00:00-04:00",  "endDt": "2014-11-27T00:00:00-05:00",  "lastUpdatedDt": "2014-11-27T07:17:00-05:00",  "meterInfoList": {  "startDt": "2014-10-28T00:00:00-04:00",  "endDt": "2014-11-27T00:00:00-05:00",  "typeCd": "unlimited",  "downloadUsedAmt": 238.4,  "uploadUsedAmt": 19.22,  "totalUsedAmt": 257.62,  "unitOfMeasureCd": "GB",  "totalChargedAmt": 0.0,  "totalChargedAmtTxt": "$0.00",  "planInfo": {  "planTxt": "High Speed Extreme (V3)",  "planIncludeAmt": 0,  "unitOfMeasureCd": "GB",  "addOnList": {  "addOnTxt": "Unlimited Internet Usage- with 3 TELUS services",  "addOnAmt": 0.0,  "unlimitedInd": true  }  },  "usageDetailList": [  {  "usageDt": "2014-11-28T01:00:01-05:00",  "uploadAmt": 551556.0,  "downloadAmt": 1.2683373E7,  … |
| **SLA** | |  |  | | --- | --- | | **Service/operation** | **SLA** | | **WirelineUsageService 1.0**  getHighSpeedUsage | 1000 ms average; 3000 ms peak | |
| **Sample GUI** |  |

## service-types

|  |  |
| --- | --- |
| **OPERATION**  usage/service-types | |
| **Method** | GET |
| **Description** | List of usage service type codes/descriptions for data and messaging display and for building a drop-down "filter" list. |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | language | "en", "fr" | Optional | Language of service types. Default is "en". | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | CustomerRatingInquirySvc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “serviceTypesResponse” {  “dataTypeList”: serviceType [],  “messageTypeList: serviceType [],  "status" : <status>},  where serviceType is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | serviceTypeCd | string | event type code | “wisp”, “wwb”, “ringtone”, “textmsg” | | serviceTypeTxt | string | Description in specified language | “data usage”, “messaging”, “downloads” |   Example:  {  "serviceTypesResponse": {  "dataTypeList": [  {  "serviceTypeCd": "wisp",  "serviceTypeTxt": "Data usage"  }, {  "serviceTypeCd": "download",  "serviceTypeTxt": "Song"  }  ],  "messageTypeList": {[  "serviceTypeCd": "textmsg",  "serviceTypeTxt": "Text messages"  ]}  ,  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **CustomerRatingInquiryService v 3.0**  getServiceTypeList | 2000 ms average; 5000 ms peak |   Expected average response time: 250 ms – this method returns reference data which should be cached and hence fast |
| **Sample GUI** |  |

## summary

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/usage/summary | |
| **Method** | GET |
| **Description** | usage summary for data or messaging (totals by service type) |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | type | "data", "messaging" | Mandatory | Specifies which type of usage to return | | from | e.g. "2013-06-01" | Mandatory | Start date of range | | to | e.g. "2013-06-30" | Optional | End date of range; if not provided default to today | | billingtype | "postpaid", "prepaid" | Optional | Required by downstream service; if provided, will avoid extra call to determine account type/subtype | | servicetype | "wisp", "wwb", "textmsg"… | Optional | Used to return a list filtered for the specified servicetype | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN | | 400 |  | Invalid date "xxxx- xx-xx" |  |  | Invalid date | | 400 |  | Invalid type parameter – must be 'data' or 'messaging' |  |  |  | | 500 |  | Summary DSAL svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | {  "summaryResponse": {  "usageSummary": {  "totalChargedAmt" : <total amount>,  "summaryList": [usageSummaryItem]  },  "status" : <status>  }  where **totalChargedAmt** is the sum of the amounts from each row, and  where **usageSummaryItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | serviceTypeCd | string | event type code | “wisp”, “wwb”, “ringtone”, “textmsg”… | | serviceTypeTxt | string | service type | e.g. “Data Usage” | | usageAmt | double | amount used | e.g. 11.31, 1.0 | | unitOfMeasureCd | string | Unit code | e.g."MB" | | unitOfMeasureTxt | string | description of unit | e.g. “Megabyte (Data Volume)” | | chargedAmt | double | amount charged | e.g. 0.00, 3.75 |   Example:  {  "summaryResponse": {  "usageSummary": {  "totalChargedAmt": 5.74,  "summaryList": [  {  "serviceTypeCd": "wisp",  "serviceTypeTxt": "Data Usage",  "usageAmt": 11.35,  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "Megabyte (Data Volume)",  "chargedAmt": 3.75  }, {  "serviceTypeCd": "download",  "serviceTypeTxt": "Music Download",  "usageAmt": 1.0,  "unitOfMeasureTxt": "song",  "chargedAmt": 2.99  }  ]  },  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SummaryDataServicesUsageService v 5.0**  getRatedUsageByServiceType | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## detail

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/usage/detail | |
| **Method** | GET |
| **Description** | usage detail for data or messaging – itemized list – one entry per day per service type |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | type | "data", "messaging" | Mandatory | Specifies which type of usage to return | | from | e.g. "2013-06-01" | Mandatory | Start date of range | | to | e.g. "2013-06-30" | Optional | End date of range; if not provided default to today | | billingtype | "postpaid", "prepaid" | Optional | Required by downstream service; if provided, will avoid extra call to determine account type/subtype | | servicetype | "wisp", "wwb", "textmsg"… | Optional | Used to return a list filtered for the specified servicetype | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN (AIS call failed) | underlying error code | underlying error message | AIS fails due to invalid BAN : BAN [90653242] Not Found  messageId=APP10004, errorCode=CMB\_AIS\_0001 | | 400 |  | Invalid date "xxxx- xx-xx" |  |  | Invalid date | | 500 |  | Summary DSAL svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "detailResponse": {  "usageDetail": {  "detailList": [usageSummaryItem]  }  },  "status" : <status>  See previous operation for description of **usageSummaryItem**. In this operation, the usageDt field will be populated.  Example:  {"detailResponse":  {  "detailList": [  {  "usageDt" : “2013-06-05T00:00:00”,  "serviceTypeCd" : “wisp”,  "serviceTypeTxt" : “Data Usage”,  "usageAmt" : 11.35,  "unitOfMeasureCd" : "MB",  “unitOfMeasureTxt” : “Megabyte (Data Volume)”,  “chargedAmt” : 3.75  },  {  “usageDt” : “2013-06-05T00:00:00”,  “serviceTypeCd” : “download”,  “serviceTypeTxt” : “Music Download”,  “usageAmt” : 1.0,  "unitOfMeasureCd" : "song",  “unitOfMeasureTxt” : “song”,  “chargedAmt” : 2.99  }  ]  },  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SummaryDataServicesUsageService v 5.0**  getDailyRatedUsage | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## usage

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/usage | |
| **Method** | GET |
| **Description** | usage summary and detail for data or messaging – combination of output from previous two operations |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | type | "data", "messaging" | Mandatory | Specifies which type of usage to return | | from | e.g. "2013-06-01" | Mandatory | Start date of range | | to | e.g. "2013-06-30" | Optional | End date of range; if not provided default to today | | billingtype | "postpaid", "prepaid" | Optional | Required by downstream service; if provided, will avoid extra call to determine account type/subtype | | servicetype | "wisp", "wwb", "textmsg"… | Optional | Used to return a list filtered for the specified servicetype | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IB | Invalid BAN (AIS call failed) | underlying error code | underlying error message | AIS fails due to invalid BAN: BAN [90653242] Not Found;  messageId=APP10004, errorCode=CMB\_AIS\_0001 | | 400 |  | Invalid date "xxxx- xx-xx" |  |  | Invalid date | | 500 |  | Summary DSAL svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | {  "summaryDetailResponse":  {  " usageSummary": {  "totalChargedAmt" : <total amount>,  "summaryList": usageSummaryItem []  },  " usageDetail": {  " detailList": usageSummaryItem []  },  "status" : <status>    See previous operations for description of **usageSummaryItem**.  Example:  “summary” {  "totalChargedAmt" : 5.74,  “summaryList”: [  {  “serviceTypeCd” : “wisp”,  “serviceTypeTxt” : “Data Usage”,  “usageAmt” : 11.35,  "unitOfMeasureCd" : "MB",  “unitOfMeasureTxt” : “Megabyte (Data Volume)”,  “chargedAmt” : 3.75  }  {  "usageDt" : null,  “serviceTypeCd” : “download”,  “serviceTypeTxt” : “Music Download”,  “usageAmt” : 1.0,  “unitOfMeasureTxt” : “song”,  “chargedAmt” : 2.99  }  ]  },  “detail” {  “detailList”: [  {  "usageDt" : “2013-06-05T00:00:00”,  "serviceTypeCd" : “wisp”,  "serviceTypeTxt" : “Data Usage”,  "usageAmt" : 11.35,  "unitOfMeasureCd" : "MB",  “unitOfMeasureTxt” : “Megabyte (Data Volume)”,  “chargedAmt” : 3.75  }  {  “usageDt” : “2013-06-05T00:00:00”,  “serviceTypeCd” : “download”,  “serviceTypeTxt” : “Music Download”,  “usageAmt” : 1.0,  "unitOfMeasureCd" : "song",  “unitOfMeasureTxt” : “song”,  “chargedAmt” : 2.99  }  ]  },  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SummaryDataServicesUsageService v 5.0**  getDailyRatedUsage  getRatedUsageByServiceType | 2000 ms average; 5000 ms peak  2000 ms average; 5000 ms peak |   Expected average response time: 4500 ms |
| **Sample GUI** |  |

## transactions

Changes from previous version highlighted in yellow.

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/usage/transactions | |
| **Method** | GET |
| **Description** | Individual transactions for a given day and service type – typically drilling down from the usage detail |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | date | e.g. "2013-06-01" | Mandatory | Date of transactions | | servicetype | "wisp", "wwb", "textmsg"… | Mandatory | Service type of transactions | | billingtype | "postpaid", "prepaid" | Optional | Required by downstream service; if provided, will avoid extra call to determine account type/subtype | | periodthresholds | 1. comma-separated list of numbers from 0-23; OR 2. 24 | Optional | 1. If provided, the service will return rolled totals for each period of the day; e.g. if "12,18" is passed as a value, we will return the total MB (or msg) for midnight to noon (12:00), noon to 6 p.m. (18:00), and 6 p.m. to midnight 2. If 24, then it will return totals for each hour, i.e. 24 values   **NOTE: this parameter will be ignored if servicetype is anything other than "wisp".** e.g. you can't get hourly breakdowns for messaging. | | totalsonly | "y", "n" | Optional | If is "y" (yes) and period-thresholds provided, do not populate transactionList (makes for more compact response) | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid/missing date "xxxx- xx-xx" |  |  | Invalid date or date not passed | | 400 |  | Missing servicetype |  |  | Servicetype not passed | | 400 |  | Invalid period-thresholds |  |  | numbers aren't 0-23 | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 500 |  | DSAL svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "transactions" {  "transactionList": transactionItem [],  "vendorCodeList": String[],  "periodTotals" : [double]  },  "status" : <status>  where **transactionItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | eventDt | Date | Time of event | 2013-06-13T00:00:00 | | eventTimeTxt | string | Localized time | 1:17 PM | | unitOfMeasureCd | string | Unit | "MB", "msg" | | countAmt | double | Amount used | e.g. 11.3, 1.0 | | chargedAmt | double | Dollar amount | 1.25 | | chargedTxt | String | Localize dollar amount | "$1.25", "1,25 $" | | networkTxt | string | Network (English or French desc) | "LTE" | | vendorCd | string | Vendor code | e.g. "Lavalife" | | eventNameTxt | string | Description of event in specified language | e.g. “Data Usage", "Text Msg – Sent" | | countryTxt | string | Source or destination | e.g. "U.S." | | recipientAddressTxt | string | Typically phone number | 4165603444 | | zoneCd | String | Zone code | CAN, US\_roam. International\_roam | | ZoneTxt | String | Localized string value for the zoneCd |  | | rlhInd | boolean | Roam Like Home indicator | true, false – indicates whether the usage was roaming rated as domestic under the RLH scheme | | mccMncCd | String | Mobile Carrier Code/ Mobile Network Code | string of 6 digits uniquely identifying a mobile phone operator. E.g.  310004 – Verizon (US)  310026 – T-Mobile  310030 – AT&T (US)  A future version of this service will return the full description as well. |   and **vendorCodeList** is a list of String elements.  Example:  "transactionsResponse" {  "transactionList" :  [  {  "eventDt" : "2013-11-05T11:52:00-05:00",  “eventTimeTxt” : “11:52:00 AM”,  “unitOfMeasureCd” : “Data Usage”,  “countAmt” : 11.35,  “chargedAmt” : “Megabyte (Data Volume)”,  “chargedTxt” : 3.75,  "networkTxt" : "LTE",  "vendorCd" : "LAVALIFE",  "eventNameTxt" : null,  "countryTxt" : null,  "recipientAddressTxt" : null,  "zoneCd":"domestic",  "zoneTxt": "Domestic",  "rlhInd" : false,  "mccMcnCd" : null  },  {  "eventDt" : "2013-11-09T10:00:00-05:00",  “eventTimeTxt” : “10:00:00 AM”,  “unitOfMeasureCd” : “Data Usage”,  “countAmt” : 11.35,  “chargedAmt” : “Megabyte (Data Volume)”,  “chargedTxt” : 3.75,  "networkTxt" : "LTE",  "vendorCd" : "",  "eventNameTxt" : null,  "countryTxt" : null,  "recipientAddressTxt" : null,  "zoneCd":"US\_roam",  "zoneTxt": "Roaming",  "rlhInd" : true,  "mccMcnCd" : "310990"  }  ],  "vendorCodeList": ["BOKU","LAVALIFE"],  "periodTotals" : [27.3, 40.2, 8.7],  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  }  }  If totals-only = "y", this would be simplified to:  “transactions” {  "transactionList" : null,  "period-totals" : [27.3, 40.2, 8.7],  "status" : …  If periodthresholds=24, periodTotals would look like this:  "periodTotals" : [0.0, 0.0, 0.5, … 27.3, 40.2, 8.7… 1.5], (24-value CSV list) |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **DataServicesUsageService v 4.3**  getBIllableDataUsage  **ProgramInformationService v 1.0**  getProgramContact | 2000 ms average; 3000 ms peak  4000 ms average; 5000 ms peak |   Expected average response time: 6000 ms (in practice, we see much lower times) |
| **Sample GUI** |  |

## refunds

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/refunds | |
| **Method** | GET |
| **Description** | Refund history with partner info |
| **Input** | **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | from | e.g. "2013-06-01" | Mandatory |  | | to | e.g. "2013-06-01" | Mandatory |  | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 200 |  | Refund svc succeeded, Partner svc failed |  |  | As long as Refund svc is successful, we return refund info but unpopulated partner info | | 400 |  | Invalid/missing date "xxxx- xx-xx" |  |  | Invalid date or date not passed | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 500 |  | SDF Refund svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “refunds” {  “refundList”: refundItem [],  "partnerList": partner[]  },  "status" : <status>  where **refundItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | refundDt | Date | Date of refund | 2013-06-13T00:00:00 | | refundDateTxt | string | Localized date | June 13, 2013 | | vendorCodeTxt | string | Vendor code | "Lavalife", "Telus" | | serviceNameTxt | string | Service name |  | | refundAmt | double | Dollar amount | 1.25 | | refundAmountTxt | String | Localized dollar amount | "$1.25", "1,25 $" |   And **partner** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | vendorCodeTxt | string | Vendor code | "Lavalife" | | nameTxt | string | Partner name | "Lavalife Dating Services" | | urlTxt | string | Partner URL | http://www.lavalife.com/support | | phoneTxt | string | 10-digit unformatted phone no. | 8003334326 | | emailTxt | string | Email address | support@lavalife.com |   Example:  “refunds” {  "refundList" : {  [  {"refundDt" : "2013-06-20T00:00:00",  “refundDateTxt” : “20 Juin 2013”,  “vendorCodeTxt” : "Lavalife",  “serviceNameTxt” : "Lavalife Match",  “refundAmt” : -15,  “refundAmountTxt” : "- 15,00 $"  },  {"refundDt" : "2013-06-13T00:00:00",  “refundDateTxt” : “13 Juin 2013”,  “vendorCodeTxt” : "MSN",  “serviceNameTxt” : "Messagerie MSN",  “refundAmt” : -2,  “refundAmountTxt” : "- 2,00 $"  }  ]  },  "partnerList" : {  [  {"vendorCodeTxt" : "lavalife",  "nameTxt" : "Lavalife Dating Services",  "urlTxt" : "http://www.lavalife.com/support",  "phoneTxt" : "4164445463",  "emailTxt" : "support@lavalife.com"  },  {"vendorCodeTxt" : "Microsoft",  "nameTxt" : "Microsoft Netowkr",  "urlTxt" : "http://www.msn.com/support",  "phoneTxt" : "8004445463",  "emailTxt" : "support@msn.com"  }  ]  },  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SDFRefundRetrievalService v 1.0**  getSubscriberRefundList  **ProgramInformationService v 1.0**  getProgramContact | 4000 ms average; 5000 ms peak  4000 ms average; 5000 ms peak |   Expected average response time: 8000 ms (in practice, we see much lower times) |
| **Sample GUI** |  |

## partners

|  |  |
| --- | --- |
| **OPERATION**  partners | |
| **Method** | GET |
| **Description** | retrieve information about 3rd party vendors ("partners") associated with data usage |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | vendorcodelist | ["BOKU", "LAVALIFE"] | Mandatory | This list would be compiled from the response to transactions. | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | invalid vendor list |  |  | unable to parse the comma-separated list of vendor codes | | 500 |  | Partner svc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “partnersResponse” {  "partnerList": partner[],  "status" : <status>  }  Where **partner** is as described above under Refunds.  Example:  “partnersResponse” {  "partnerList" : {  [  {"vendorCodeTxt" : "LAVALIFE",  "nameTxt" : "Lavalife Dating Services",  "urlTxt" : "http://www.lavalife.com/support",  "phoneTxt" : "4164445463",  "emailTxt" : "support@lavalife.com"  },  {"vendorCodeTxt" : "MICROSOFT",  "nameTxt" : "Microsoft Network",  "urlTxt" : "http://www.msn.com/support",  "phoneTxt" : "8004445463",  "emailTxt" : "support@msn.com"  }  ]  },  "status" :  { "statusCd" : "200",  "statusTxt" : "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **ProgramInformationService v 1.0**  getProgramContact | 1000 ms average; 2000 ms peak |   Expected average response time: 1000 ms |
| **Sample GUI** |  |

## top-usage

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/top-usage | |
| **Method** | GET |
| **Description** | Phone numbers called and texted most often in previous months |
| **Input** | **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | subscriptionid | Subscription ID | Optional | When provided, implementation can bypass call to SubscriberInfoService. | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | TBD |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 500 |  | usage svc failed |  |  | Any Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “localVoiceUsageList” : [<voiceUsage]>,  “longDistanceVoiceUsageList” : [<voiceUsage]>,  “messageUsageList” : [<messageUsage]>,  "status" : <status>  where **voiceUsage** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | phoneNumberTxt | string | Phone number | "4164657887" | | totalIncomingMinutesNum | integer | Total incoming minutes for the given phone number | 17 | | totalOutgoingMinutesNum | integer | Total outgoing minutes for the given phone number | 23 |   And **messageUsage** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | phoneNumberTxt | string | Phone number | "4164657887" | | totalMessagesReceivedNum | integer | Total incoming messages for the given phone number | 17 | | totalMessagesSentNum | integer | Total outgoing messages for the given phone number | 23 |   Example:  {  "localVoiceUsageList": [  {  "phoneNumberTxt": "41644439807",  "totalIncomingMinutesNum": 12,  "totalOutgoingMinutesNum": 23  }, {  "phoneNumberTxt": "41644439666",  "totalIncomingMinutesNum": 53,  "totalOutgoingMinutesNum": 534  }  ],  "longDistanceVoiceUsageList": [  {  "phoneNumberTxt": "61344439807",  "totalIncomingMinutesNum": 12,  "totalOutgoingMinutesNum": 23  }, {  "phoneNumberTxt": "61344439666",  "totalIncomingMinutesNum": 53,  "totalOutgoingMinutesNum": 534  }  ],  "messageUsageList": [  {  "phoneNumberTxt": "41644439807",  "totalMessagesReceivedNum": 12,  "totalMessagesSentNum": 23  }, {  "phoneNumberTxt": "41644439666",  "totalMessagesReceivedNum": 53,  "totalMessagesSentNum": 534  }  ]  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **UsageSummaryService v 1.0**  findTopVoiceNumber  findTopPhoneNumsForMessaging | 500 ms average; 1000 ms peak  500 ms average; 1000 ms peak | | **SubscriberInformationService 3.0**  getSubscriberIdentifierByPhoneNumberAndAccountNumber | 2000 ms average; 5000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## prepaid-meters

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/prepaid-meters | |
| **Method** | GET |
| **Description** | Data, messaging and voice usage meters by add-on |
| **Input** | **ban** – account number (prepaid only)  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated. | | language | "en", "fr" | Optional | Language of decoration. Default is "en". |   Example:  UsageManagement/account/2343465/sub/4165765687/prepaid-meters |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 500 |  | SubscriptionManagementSvc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | ProductOfferingSvc failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  "meterList" : [<PrepaidMeter>],  "status” : <status>  }  where < PrepaidMeter > is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterTypeCd | string | type of meter | “data”, “voice”, “messaging” |  | | serviceCd | string | feature code that meter is attached to | “514” |  | | serviceTxt | string | localized description of feature | “Data 10”, “Messaging 250” |  | | usedAmt | int | amount used from allowance | 34 |  | | remainingAmt | int | total amount used (including free) | 35 |  | | unitOfMeasureCd | string | unit of measure | “min”, “mb”, “msg” |  | | serviceExpiryDt | date | expiry date of feature | "2015-03-20T01:00:00-0400" |  |   Example:  {  "meterList": [  {  "meterTypeCd": “data”,  "serviceCd": 413,  "serviceTxt": "Data 10",  "usedAmt": "37.0",  "remainingAmt": "63.0",  "unitOfMeasureCd": "mb",  "serviceExpiryDt": "2015-05-03T00:00:00-0400"  }, {  "meterTypeCd": “voice”,  "serviceCd": 547,  "serviceTxt": "Unlimited Talk + Messaging 55",  "usedAmt": "253.0",  "remainingAmt": "unlimited",  "unitOfMeasureCd": "min",  "serviceExpiryDt": "2015-05-03T00:00:00-0400"  }, {  "meterTypeCd": “messaging”,  "serviceCd": 547,  "serviceTxt": "Unlimited Talk + Messaging 55",  "usedAmt": "17",  "remainingAmt": "unlimited",  "unitOfMeasureCd": "msg",  "serviceExpiryDt": "2015-05-03T00:00:00-0400"  }  ]  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SubscriptionService 2.1**  getSubscribedProducts  **ProductOfferingService**  getProductOfferings |  |   Expected average response time: 1000 ms |
| **Sample GUI** |  |

# UsageMeter REST Service

This service provides operations in support of displaying data and messaging usage meters. It is designed to replace data-and-messaging-meters from UsageManagement service, and will supersede it once the new "Keystone" rating engine goes live in October 2014. We will move the two operations in this service into UsageManagement in the near future.

## detailed-data-and-messaging-meters

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/detailed-data-and-messaging-meters | |
| **Method** | GET |
| **Description** | Detailed Data and Messaging meters for current bill cycle and pass five cycles. |
| **Input** | **ban –** account number (currently only wireless supported; postpaid only}  **sub –** subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | subscriberid | Subscriber ID | Optional | When provided, implementation can bypass call to SubscriberInfoService. | | billcycle | Bill cycle number | Optional | When these are both provided, implementation can bypass call to AccountInformationService. It can be obtained by the initial call to bill-cycle-info. | | enddate | Bill cycle end date  yyyy-mm-dd | | cycleperiod | which bill cycle to return – current and past five | Optional | If the month that is passed in is current billing cycle month, then the value is 0, if it is last month then ---1, etc. If not provided, the parameter defaults to the value of 0.  current usage cycle 0  last month -1  two months ago -2  three months ago -3  four months ago -4  five months ago -5 | | serviced |  | Optional | For future use. | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="n"), response will not be decorated. | | language | "en", "fr" | Optional | Language of decoration. Default is "en". If decorate parameter is missing, this will be ignored. | | socdetails | “y”, “n” | Optional | service and plan descriptions will be provided if “y” is specified, service and plan description will bypassed and not provided if “n” is specfied (Default is y) introduced in minor v 1.1 | | appid | Indicates which set of resources to decorate with | Optional | "web-telus", "device-telus", "web-koodo", "device-koodo" (NOT implemented) | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | **Notes** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN (AIS call failed) | underlying error code | underlying error message | AIS fails due to invalid BAN – AIS errors:  BAN [90653242] Not Found  messageId=APP10004, errorCode=CMB\_AIS\_0001 | | 400 | IS | Invalid Sub (SIS call failed) | underlying error code | underlying error message | SIS fails due to invalid sub – SIS errors: Subscriber not found for phone number: 4165765688;  messageId= APP20002,  errorCode=CMB\_SIS\_0015 | | 400 | IBC | Invalid bill cycle (CRI call failed) | underlying error code | underlying error message | CRI invalid bill cycle. Include error msg which indicates correct bill cycle: CRI-000016: "Invalid bill cycle"; messageId= CRI-000016,  errorCode= CRI-000016,  errorMessage=xxxx | | 500 |  | CustomerRatingInquiryService call failed | underlying error code | underlying error message | Any other CRI exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | {  "getDetailedDataAndMessagingMetersResponse": {  "data": {  "meterGroupZoneList": [  {  "zoneCd": "domestic",  "zoneTxt": "Domestic",  "meterGroupList": [  {  "groupNameCd": "domesticFixed",  "groupNameTxt": "Domestic Fixed",  "totalUsedAmt": 3696,  "totalIncludedAmt": 5720,  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "meterList": [  {  "meterHeadingCd": "recurringAllowances",  "meterHeadingTxt": "Recurring Allowances",  "typeCd": "allowance",  "meterNum": 1,  "serviceCd": "SX34222",  "serviceNameTxt": "Data Plan #1",  "serviceEffectiveDateTxt": "2013-05-22T05:07:23Z",  "servicExpiryDateTxt": "2013-06-22T05:07:23Z",  "usedAmt": 512,  "includedAmt": 1024,  "chargedAmt": 40.0,  "chargedTxt": "$40.00",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "statusCd": "active",  "sharedServiceInd": false,  "sharingGroupCd": "",  “sharingGroupNameTxt": "",  “sharingAddOnInd": "",  "unlimitedInd": "false",  "pricePlanInd": "false"  }, ...]  }  ]  }, {  "groupNameCd": "domesticShare",  "groupNameTxt": "Domestic Share",  "totalUsedAmt": 730,  "totalIncludedAmt": 2048,  ...  "meterList": [  ...  ]  }, {  "groupNameCd": "domesticBCIC",  ...  }, {  "zoneCd": "roaming",  "zoneTxt": "Roaming",  "meterGroupList": [  {  "groupNameCd": "roamingFixed",  ...    "totalByZoneList": [  {  "zoneCd": "domestic",  "zoneTxt": "Domestic Data",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "Mo",  "totalUsageAmt": 5890,  "inAllowanceUsageAmt": 5120,  "flexChargedAmt": 25.26,  "flexChargedTxt": "$25.26",  "overageUsageAmt": 770,  "overageUsageChargedAmt": 15.40,  "overageUsageChargedTxt": "$15.40"  }, {  "zoneCd": "roaming",  "zoneTxt": "Roaming Data",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "Mo",  "totalUsageAmt": 2160,  "inAllowanceUsageAmt": 2048,  "flexChargedAmt": 0,  "flexChargedTxt": "$0.00",  "overageUsageAmt": 112,  "overageUsageChargedAmt": 46.00,  "overageUsageChargedTxt": "$46.00"  }  ]  },  "messaging": {  "meterGroupZoneList": [  {  ...  "totalByZoneList": [  ... ]  },  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }  }  **Full Response:**    Meters within the data and message section are group by zone (domestic or roaming) and then by meter categories (e.g Domestic Fixed, Domestic etc ).  The detailed-data-and-message operation, will return the usage at the service (SOC level), whereas in the data-and-message meters operation returns the aggregated usage at the service type level.  Note: Any fields with empty values are not returned.  **DetailedDataAndMessagingMeters**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | data | data object | The usage is segregated into two sections: data and message. This section represents the data | Not returned if no data usage. |  | | messaging | message object | The usage is segregated into two sections: data and message. This section represents the message | Not returned if no messaging usage. |  | | subscriberId | Subscriber ID | Optional | Returned if not provided in input |  | | billcycle | Bill cycle number | Optional | Returned if not provided in input |  | | endDt | Bill cycle end date  yyyy-mm-dd | Optional | Returned if not provided in input |  |   **Data/Message**  Data and message have a common structure.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterGroupZoneList | List of meterGroupZone | Holds the usage information within the data or Message elements grouped by zones and meterGroups |  |  | | totalChargedAmt | double | chargeAmts in each meterGroups aggregated to either data or message level in dollars. |  |  | | totalChargedTxt | String | Localized totalChargeAmt string |  |  | | totalByZoneList | List of totalByZone | Totals in each meterGroup aggregated to zones within the data or message element. |  |  |   **TotalByZone**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | zoneCd | string | Zone code as returned by the CRI |  |  | | zoneTxt | string | Localized zone name | Domestic/Roaming |  | | unitOfMeasureCd | string | Unit of measures for the aggregated usage elements the totalsByZone | “MB” |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. | “MB” |  | | totalUsageAmt | double | Aggregation of total usage amount in each meterGroups under data or message element grouped by zone. (inAllowanceUsageAmt + overageUsageAmt) |  |  | | inAllowanceUsageAmt | double | Aggregation of the total allowance. : All allowance. (- including shareable and non-shareable -Rate Plans/Add-Ons, BCICs, BCDCs, Boosters, Flex Plans. (in MB unit of measure)  Note: Flex allowance are the sum of the allowance in each tier up to the current tier (including the current tier). The current tier is determined as follows:   * + 1. If no usage, the current tier is the first tier.     2. If any allowance tier is partially consumed, it is the current tier.     3. If any tier is fully consumed and the following tier not consumed, the following tier is the current tier. |  |  | | flexChargedAmt | double | Charges associated with crossing flex tiers (in $ unit of measure). Flex charges are aggregated up to the consumed tiers. Excludes PPU charges. |  |  | | flexChargedTxt | String | Localized dollar string of flexChargesAmt |  |  | | overageUsageAmt | double | Aggregation of usage amount in PPU meters (type =ppu) and the PPU tier of the tiered meters. |  |  | | overageUsageChargedAmt | double | Aggregation of charges in dollars in PPU meters (type =ppu) and the PPU tier of the tiered meters. |  |  | | overageUsageChargedTxt | String | Localized string of overageUsageCharge |  |  |   **meterGroupZone**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | zoneCd | string | Zone code as returned by CRI |  |  | | zoneTxt | string | Localized zone name | Domestic/Roaming |  | | meterGroupList | List of meterGroup | List of meter Groups.  The PPU meters are aggregated per zones. If there is usage, the following PPU meters may be returned:   1. Domestic Data PPU meter 2. Roaming Data PPU meter 3. Domestic Messaging PPU meter 4. Roaming Messaging PPU meter.   Meters without any groupings in SAPCC (besides PPU), will come under the “Other” group. |  |  |   **meterGroup**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typicl values** | **Decoration** | | groupNameCd | string | Code for the meter group |  |  | | groupNameTxt | string | Localized name associated with the nameCd. | Domestic:  Domestic Fixed, Domestic Share, Domestic BCIC, Domestic BCIC Share, Domestic Data Boosters, Domestic Data Boosters Shared, PPU, Other  Roaming:  Roaming Fixed, Roaming Share, Roaming BCIC, Roaming BCIC Share, Roaming Data Boosters, Roaming Data Boosters, PPU, Other |  | | totalUsedAmt | double | Aggregated usage amount for the meters within the group. This does not include the PPU tier usage from the Flex meter. | 2023.3 |  | | totalIncludedAmt | double | Aggregated allowance for the meters within the group | 2408 |  | | unitOfMeasureCd | string | Unit of measures for the totalUsedAmt and totalIncludedAmt at the group level | “MB” |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. | “MB” |  | | meterList | array of meter | List of service (SOC) level meters |  |  |   **Meter**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterHeadingCd | String | meterHeadingCd returned by CRI. Could be empty |  |  | | meterHeadingTxt | string | Localized description for the meterHeadingCd | Recurring Allowances, 1-time data boosters. Flex Plans or empty string |  | | typeCd | string | meter type | "allowance", "unlimited", "ppu", "tier", "share", "datasaver" |  | | meterNum | integer | ordinal | e.g. 1, 2, 3… |  | | serviceCd | String | Service code from KB that the meter is associated with. | e.g. "SXCC10C", etc.  Irrelevant for PPU meter |  | | serviceNameTxt | string | Service code description | "2GB Data"  This field is irrelevant for PPU meter |  | | serviceEffectiveDt | Date String | ISO8601 format date/time | "2013-05-22T05:07:22Z"  Irrelevant for PPU meter |  | | serviceExpiryDt | Date String | ISO8601 format date/time | "2013-05-22T05:07:22Z"  Irrelevant for PPU meter |  | | usedAmt | double | consumed amount | e.g. 512.0 (MB), 23.0 (messages) |  | | includedAmt | double | allowance | e.g. 2048.0 MB, 100.0 messages |  | | chargedAmt | double | Dollar amount charged. This does not include the PPU tier charge from the Flex meter. | e.g. 2.40 |  | | chargedTxt | String | Localized dollar amount chargedAmt. | $2.40 |  | | unitOfMeasureCd | String | Unit of measure for the usedAmt and includedAmt | “MB” |  | | unitOfMeasureTxt | String | Localized string for unitOfMeasure | “MB” |  | | statusCd | string | active vs. expired | "active", "expired" |  | | sharedServiceInd | boolean | Indicates if the SOC associated with this meter is shared among multiple subscribers | “true”, “false” |  | | sharingGroupCd | String | sharingGroupCd returned by the CRI. Required for the follow-up call to get the shared usage  Only present when the SOC belongs to the meter is shared. |  |  | | sharingGroupPhoneNumberList | List of stings | List of phone numbers participating in the shared service  Only present when the SOC belongs to the meter is shared. |  |  | | sharingGroupNameTxt | String | Description for the sharing group name. The description associated with the sharingGroupCd  (since v1.1) |  |  | | sharingAddOnInd | boolean | True if the meter is for sharing add-ons. BCIC and BCDC. | “true”, “false” |  | | tierInfoList | List of tier info | List of tiers for the flex add-on  Only present when the SOC belongs to the meter is flex. |  |  | | pricePlanInd | boolean | Indicates if the usage meter belongs to a price plan or add-on service. Returns true if the SOC is price plan otherwise returns false for Add-Ons. | “true” or “false” |  | | unlimitedInd | boolean | Indicates if the meter has unlimited allowance | “true” or “false” |  |   **tierInfo**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | tierNum | integer | tier number | e.g. 0, 1… |  | | tierAllowanceAmt | decimal | Allowance for each tier | e.g. 1, 2, 3… |  | | tierConsumptionAmt | double | consumed amount for each tier | e.g. 512.0 (MB), 23.0 (messages) |  | | chargedAmt | double | Charged amount for the tier |  |  | | chargedTxt | string | Localized dollar value of chargedAmt |  |  | | tierLowerLimitNum | decimal | Lower limit of the tier | e.g for a tier goes from 1024 MB to 2048, lower limit is 1024 |  | | tierUpperLimitNum | decimal | Upper limit of the tier | e.g for a tier goes from 1024 MB to 2048, upper limit is 2048 |  | | typeCd | String | Type of the tier: allowance or ppu | Allowance/ppu |  | |
|  |  |

## shared-data-and-messaging-usage

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/shared-data-and-messaging-usage | |
| **Method** | GET |
| **Description** | Returns the shared usage at subscriber level for the shared meters. |
| **Input** | **ban –** account number (currently only wireless supported; postpaid only}  **sub –** subscriber phone number  Parameters:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | subscriberid | Subscriber ID | Optional | When provided, implementation can bypass call to SubscriberInfoService. | | billCycle | Bill cycle number | Optional | When these are both provided, implementation can bypass call to AccountInformationService. It can be obtained by the initial call to bill-cycle-info. | | Enddate | Bill cycle end date  yyyy-mm-dd | | cycleperiod | Cycle that the shared usage belongs to. | Optional | If the month that is passed in is current billing cycle month, then the value is 0, if it is last month then ---1, etc. Defaults to 0 if not provided  current usage cycle 0  last month -1  two months ago -2  three months ago -3  four months ago -4  five months ago -5 | | language | "en", "fr" | Optional | Language of decoration. Default is "en". If decorate parameter is missing, this will be ignored. | | Appid | Indicates which set of resources to decorate with | Optional | "web-telus", "device-telus", "web-koodo", "device-koodo" (NOT implemented) | | serviceId |  | Optional | For future use. | | sharinggroupcd | Returned by the shared detailed-data-and-messaging-meters | Mandatory |  | | sharingPhoneNumberList | List of shared group phone numbers returned by the detailed-data-and-messaging-meters | Mandatory | sharing  PhoneNumberList=  {"phoneNumber":  ["4165552314",  "9058331234"]} | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | **Notes** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN (AIS call failed) | underlying error code | underlying error message | AIS fails due to invalid BAN – AIS errors:  BAN [90653242] Not Found  messageId=APP10004, errorCode=CMB\_AIS\_0001 | | 400 | IS | Invalid Sub (SIS call failed) | underlying error code | underlying error message | SIS fails due to invalid sub – SIS errors: Subscriber not found for phone number: 4165765688;  messageId= APP20002,  errorCode=CMB\_SIS\_0015 | |  |  |  |  |  |  | | 400 | IBC | Invalid bill cycle (CRI call failed) | underlying error code | underlying error message | CRI invalid bill cycle. Include error msg which indicates correct bill cycle: CRI-000016: "Invalid bill cycle"; messageId= CRI-000016,  errorCode= CRI-000016,  errorMessage=xxxx | | 500 |  | CustomerRatingInquiryService call failed | underlying error code | underlying error message | Any other CRI exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | {  " getSharedDataAndMessagingUsageResponse": {  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "sharedUsageList": [  {  "phoneNumber": "4164444444",  "usedAmt": 200  }, {  "phoneNumber": "4164444445",  "usedAmt": 150  }, {  "phoneNumber": "4164444446",  "usedAmt": 120  }, {  "phoneNumber": "4164444447",  "usedAmt": 100  }, {  "phoneNumber": "4164444448",  "usedAmt": 68  }  ]  }  }  **sharedUsageDetails**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | unitOfMeasureCd | string | Unit of measures for the usage elements within the sharedUsageList |  |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. |  |  | | sharedUsageList | List sharedUsage | List of shared usage elements. Each shared usage element in the represents data or message use by a subscriber participated in the shared meter. |  |  |   **sharedUsage**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | phoneNumber | string | Holds the usage information within the data or Message elements grouped by zones and meterGroups |  |  | | usedAmt | decimal | chargeAmts in each meterGroups aggregated to either data or message level in dollars. |  |  | |

# UsageMeter 2.0 REST Service

This service provides operations in support of displaying data and messaging usage meters. It is designed to replace data-and-messaging-meters from UsageManagement service along with Keystone rating engine data

|  |
| --- |
| **Notes for Current Consumers of Version 1**  Please note the following changes related to the fact that this is now a native REST service:   * Numerics and booleans are now unquoted (e.g. “totalUsedAmt” : **240** rather than “totalUsedAmt” : **“240”**) * The response wrapper **getDetailedDataAndMessagingMetersResponse** has been stripped away * HTTP Status code is now set to match the Status object statusCd (i.e. it won’t always be 200) |

## detailed-data-and-messaging-meters

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/detailed-data-and-messaging-meters | |
| **Method** | GET |
| **Description** | Detailed Data and Messaging meters for current bill cycle and pass five cycles. |
| **Input** | **ban –** account number (currently only wireless supported; postpaid only}  **sub –** subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | billcycle | Bill cycle number | Optional | When these are both provided, implementation can bypass call to AccountInformationService. It can be obtained by the initial call to bill-cycle-info. | | enddate | Bill cycle end date  yyyy-mm-dd | | cycleperiod | which bill cycle to return – current and past five | Optional | If the month that is passed in is current billing cycle month, then the value is 0, if it is last month then ---1, etc. If not provided, the parameter defaults to the value of 0.  current usage cycle 0  last month -1  two months ago -2  three months ago -3  four months ago -4  five months ago -5 | | serviced |  | Optional | For future use. | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="n"), response will not be decorated. | | language | "en", "fr" | Optional | Language of decoration. Default is "en". If decorate parameter is missing, this will be ignored. | | socdetails | “y”, “n” | Optional | service and plan descriptions will be provided if “y” is specified, service and plan description will bypassed and not provided if “n” is specfied (Default is y) | | sublist | “y”, “n” - flag to indicate whether to return the list of subscribers participating in the sharing group (for shared meters). Can improve performance for larger BANs. | Optional | Default is “y”.  NEW FOR NOV 2016 | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | **Notes** | | 200 |  | OK |  |  |  | | 400 |  | validation errors: {[DG\_SIS\_404 - Account number is invalid.]} |  |  |  | | 400 |  | validation errors: {[Invalid phone number for BAN - Request is invalid.]} |  |  | The phone number is not associated with the given BAN | | 400 | IBC | Invalid bill cycle (CRI call failed) | underlying error code | underlying error message | CRI invalid bill cycle. Include error msg which indicates correct bill cycle: CRI-000016: "Invalid bill cycle"; messageId= CRI-000016,  errorCode= CRI-000016,  errorMessage=xxxx | | 500 |  | CustomerRatingInquiryService call failed | underlying error code | underlying error message | Any other CRI exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | Sample Request:  /account/70744082/sub/4161514773/detailed-data-and-messaging-meters?cycleperiod=0&socdetails=y  Response:  {  "status": {  "statusCd": "200",  "statusTxt": "OK"  },  "data": {  "meterGroupZoneList": [ {  "zoneCd": "DOMESTIC",  "zoneTxt": "Domestic",  "meterGroupList": [  {  "groupNameCd": "DOMESTIC\_FIX",  "groupNameTxt": "Domestic Plans & Add-Ons",  "totalUsedAmt": 0,  "totalIncludedAmt": 0,  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "meterList": [ {  "meterHeadingCd": "RECURRING\_ALLOWANCE",  "meterHeadingTxt": "Plans & Add-ons",  "typeCd": "allowance",  "meterNum": 1,  "serviceCd": "SDT1GSHP",  "serviceNameTxt": "1GB Shareable Data",  "serviceEffectiveDt": "2016-04-20T00:00:00-0400",  "serviceExpiryDt": "2016-06-08T10:07:01-0400",  "usedAmt": 0,  "includedAmt": 1024,  "unlimitedInd": false,  "chargedAmt": 0,  "chargedTxt": "$0.00",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "statusCd": "expired",  "sharedServiceInd": true,  "sharingGroupCd": "27",  "sharingGroupPhoneNumberList": [],  "tierInfoList": [],  "pricePlanInd": false  }]  },  {  "groupNameCd": "DOMESTIC\_SHARE",  "groupNameTxt": "Domestic Share Plans & Add-ons",  "totalUsedAmt": 150,  "totalIncludedAmt": 1024,  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "meterList": [ {  "typeCd": "share",  "meterNum": 1,  "serviceCd": "SDT1GSHP",  "serviceNameTxt": "1GB Shareable Data",  "serviceEffectiveDt": "2016-06-08T00:00:00-0400",  "serviceExpiryDt": "9999-12-30T19:00:00-0500",  "usedAmt": 150,  "includedAmt": 1024,  "unlimitedInd": false,  "chargedAmt": 0,  "chargedTxt": "$0.00",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "statusCd": "active",  "sharedServiceInd": true,  "sharingGroupCd": "27",  "sharingGroupPhoneNumberList": [  "4161514773",  "4161515165"  ],  "tierInfoList": [],  "pricePlanInd": false  }]  }  ]  }],  "totalChargedAmt": 0,  "totalChargedTxt": "$0.00",  "totalByZoneList": [ {  "zoneCd": "DOMESTIC",  "zoneTxt": "Domestic",  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "totalUsageAmt": 100,  "inAllowanceUsageAmt": 0,  "flexChargedAmt": 0,  "flexChargedTxt": "$0.00",  "overageUsageAmt": 0,  "overageUsageChargedAmt": 0,  "overageUsageChargedTxt": "$0.00"  }]  },  "subscriptionId": "8382352",  "billCycle": 30,  "endDt": "2016-06-30T15:27:58-0400"  }  Meters within the data and message section are group by zone (domestic or roaming) and then by meter categories (e.g Domestic Fixed, Domestic etc ).  The detailed-data-and-message operation, will return the usage at the service (SOC level), whereas in the data-and-message meters operation returns the aggregated usage at the service type level.  Note: Any fields with empty values are not returned.  **DetailedDataAndMessagingMeters**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | data | data object | The usage is segregated into two sections: data and message. This section represents the data | Not returned if no data usage. |  | | messaging | message object | The usage is segregated into two sections: data and message. This section represents the message | Not returned if no messaging usage. |  | | billcycle | Bill cycle number | Optional | Returned if not provided in input |  | | endDt | Bill cycle end date  yyyy-mm-dd | Optional | Returned if not provided in input |  |   **Data/Message**  Data and message have a common structure.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterGroupZoneList | List of meterGroupZone | Holds the usage information within the data or Message elements grouped by zones and meterGroups |  |  | | totalChargedAmt | double | chargeAmts in each meterGroups aggregated to either data or message level in dollars. |  |  | | totalChargedTxt | String | Localized totalChargeAmt string |  |  | | totalByZoneList | List of totalByZone | Totals in each meterGroup aggregated to zones within the data or message element. |  |  |   **TotalByZone**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | zoneCd | string | Zone code as returned by the CRI |  |  | | zoneTxt | string | Localized zone name | Domestic/Roaming |  | | unitOfMeasureCd | string | Unit of measures for the aggregated usage elements the totalsByZone | “MB” |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. | “MB” |  | | totalUsageAmt | double | Aggregation of total usage amount in each meterGroups under data or message element grouped by zone. (inAllowanceUsageAmt + overageUsageAmt) |  |  | | inAllowanceUsageAmt | double | Aggregation of the total allowance. : All allowance. (- including shareable and non-shareable -Rate Plans/Add-Ons, BCICs, BCDCs, Boosters, Flex Plans. (in MB unit of measure)  Note: Flex allowance are the sum of the allowance in each tier up to the current tier (including the current tier). The current tier is determined as follows:   * + 1. If no usage, the current tier is the first tier.     2. If any allowance tier is partially consumed, it is the current tier.     3. If any tier is fully consumed and the following tier not consumed, the following tier is the current tier. |  |  | | flexChargedAmt | double | Charges associated with crossing flex tiers (in $ unit of measure). Flex charges are aggregated up to the consumed tiers. Excludes PPU charges. |  |  | | flexChargedTxt | String | Localized dollar string of flexChargesAmt |  |  | | overageUsageAmt | double | Aggregation of usage amount in PPU meters (type =ppu) and the PPU tier of the tiered meters. |  |  | | overageUsageChargedAmt | double | Aggregation of charges in dollars in PPU meters (type =ppu) and the PPU tier of the tiered meters. |  |  | | overageUsageChargedTxt | String | Localized string of overageUsageCharge |  |  |   **meterGroupZone**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | zoneCd | string | Zone code as returned by CRI |  |  | | zoneTxt | string | Localized zone name | Domestic/Roaming |  | | meterGroupList | List of meterGroup | List of meter Groups.  The PPU meters are aggregated per zones. If there is usage, the following PPU meters may be returned:   1. Domestic Data PPU meter 2. Roaming Data PPU meter 3. Domestic Messaging PPU meter 4. Roaming Messaging PPU meter.   Meters without any groupings in SAPCC (besides PPU), will come under the “Other” group. |  |  |   **meterGroup**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typicl values** | **Decoration** | | groupNameCd | string | Code for the meter group |  |  | | groupNameTxt | string | Localized name associated with the nameCd. | Domestic:  Domestic Fixed, Domestic Share, Domestic BCIC, Domestic BCIC Share, Domestic Data Boosters, Domestic Data Boosters Shared, PPU, Other  Roaming:  Roaming Fixed, Roaming Share, Roaming BCIC, Roaming BCIC Share, Roaming Data Boosters, Roaming Data Boosters, PPU, Other |  | | totalUsedAmt | double | Aggregated usage amount for the meters within the group. This does not include the PPU tier usage from the Flex meter. | 2023.3 |  | | totalIncludedAmt | double | Aggregated allowance for the meters within the group | 2408 |  | | unitOfMeasureCd | string | Unit of measures for the totalUsedAmt and totalIncludedAmt at the group level | “MB” |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. | “MB” |  | | meterList | array of meter | List of service (SOC) level meters |  |  |   **Meter**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | meterHeadingCd | String | meterHeadingCd returned by CRI. Could be empty |  |  | | meterHeadingTxt | string | Localized description for the meterHeadingCd | Recurring Allowances, 1-time data boosters. Flex Plans or empty string |  | | typeCd | string | meter type | "allowance", "unlimited", "ppu", "tier", "share", "datasaver" |  | | meterNum | integer | ordinal | e.g. 1, 2, 3… |  | | serviceCd | String | Service code from KB that the meter is associated with. | e.g. "SXCC10C", etc.  Irrelevant for PPU meter |  | | serviceNameTxt | string | Service code description | "2GB Data"  This field is irrelevant for PPU meter |  | | serviceEffectiveDt | Date String | ISO8601 format date/time | "2013-05-22T05:07:22Z"  Irrelevant for PPU meter |  | | serviceExpiryDt | Date String | ISO8601 format date/time | "2013-05-22T05:07:22Z"  Irrelevant for PPU meter |  | | usedAmt | double | consumed amount | e.g. 512.0 (MB), 23.0 (messages) |  | | includedAmt | double | allowance | e.g. 2048.0 MB, 100.0 messages |  | | chargedAmt | double | Dollar amount charged. This does not include the PPU tier charge from the Flex meter. | e.g. 2.40 |  | | chargedTxt | String | Localized dollar amount chargedAmt. | $2.40 |  | | unitOfMeasureCd | String | Unit of measure for the usedAmt and includedAmt | “MB” |  | | unitOfMeasureTxt | String | Localized string for unitOfMeasure | “MB” |  | | statusCd | string | active vs. expired | "active", "expired" |  | | sharedServiceInd | boolean | Indicates if the SOC associated with this meter is shared among multiple subscribers | “true”, “false” |  | | sharingGroupCd | String | sharingGroupCd returned by the CRI. Required for the follow-up call to get the shared usage  Only present when the SOC belongs to the meter is shared. |  |  | | sharingGroupPhoneNumberList | List of stings | List of phone numbers participating in the shared service  Only present when the SOC belongs to the meter is shared. |  |  | | sharingGroupNameTxt | String | Description for the sharing group name. The description associated with the sharingGroupCd  (since v1.1) |  |  | | sharingAddOnInd | boolean | True if the meter is for sharing add-ons. BCIC and BCDC. | “true”, “false” |  | | tierInfoList | List of tier info | List of tiers for the flex add-on  Only present when the SOC belongs to the meter is flex. |  |  | | pricePlanInd | boolean | Indicates if the usage meter belongs to a price plan or add-on service. Returns true if the SOC is price plan otherwise returns false for Add-Ons. | “true” or “false” |  | | unlimitedInd | boolean | Indicates if the meter has unlimited allowance | “true” or “false” |  |   **tierInfo**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | tierNum | integer | tier number | e.g. 0, 1… |  | | tierAllowanceAmt | decimal | Allowance for each tier | e.g. 1, 2, 3… |  | | tierConsumptionAmt | double | consumed amount for each tier | e.g. 512.0 (MB), 23.0 (messages) |  | | chargedAmt | double | Charged amount for the tier |  |  | | chargedTxt | string | Localized dollar value of chargedAmt |  |  | | tierLowerLimitNum | decimal | Lower limit of the tier | e.g for a tier goes from 1024 MB to 2048, lower limit is 1024 |  | | tierUpperLimitNum | decimal | Upper limit of the tier | e.g for a tier goes from 1024 MB to 2048, upper limit is 2048 |  | | typeCd | String | Type of the tier: allowance or ppu | Allowance/ppu |  | |
|  |  |

## shared-data-and-messaging-usage

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/shared-data-and-messaging-usage | |
| **Method** | GET |
| **Description** | Returns the shared usage at subscriber level for the shared meters. |
| **Input** | **ban –** account number (currently only wireless supported; postpaid only}  **sub –** subscriber phone number  Parameters:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Description** | **Mandatory?** | **Comments** | | subscriberid | Subscriber ID | Optional | When provided, implementation can bypass call to SubscriberInfoService. | | billcycle | Bill cycle number | Optional | When these are both provided, implementation can bypass call to AccountInformationService. It can be obtained by the initial call to bill-cycle-info. | | enddate | Bill cycle end date  yyyy-mm-dd | | cycleperiod | Cycle that the shared usage belongs to. | Optional | If the month that is passed in is current billing cycle month, then the value is 0, if it is last month then ---1, etc. Defaults to 0 if not provided  current usage cycle 0  last month -1  two months ago -2  three months ago -3  four months ago -4  five months ago -5 | | language | "en", "fr" | Optional | Language of decoration. Default is "en". If decorate parameter is missing, this will be ignored. | | appid | Indicates which set of resources to decorate with | Optional | "web-telus", "device-telus", "web-koodo", "device-koodo" (NOT implemented) | | serviceId |  | Optional | For future use. | | sharinggroupcd | Returned by the shared detailed-data-and-messaging-meters | Mandatory |  | | sharingphonenumberlist | Comma separated list of shared group phone numbers returned by the detailed-data-and-messaging-meters | Mandatory | sharingphonenumberlist=4165552314,9058331234 | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | **Notes** | | 200 |  | OK |  |  |  | | 400 |  | validation errors: {[DG\_SIS\_404 - Account number is invalid.]} |  |  |  | | 400 |  | validation errors: {[Invalid phone number for BAN - Request is invalid.]} |  |  | The phone number is not associated with the given BAN | | 400 | IBC | Invalid bill cycle (CRI call failed) | underlying error code | underlying error message | CRI invalid bill cycle. Include error msg which indicates correct bill cycle: CRI-000016: "Invalid bill cycle"; messageId= CRI-000016,  errorCode= CRI-000016,  errorMessage=xxxx | | 500 |  | CustomerRatingInquiryService call failed | underlying error code | underlying error message | Any other CRI exception | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not handled elsewhere | |
| **Output** | SampleRequest  /account/70744082/sub/4161515165/shared-data-and-messaging-usage? sharinggroupcd=27&sharingphonenumberlist=4161515165,4161514773  Response  {  "status": {  "statusCd": "200",  "statusTxt": "OK"  },  "unitOfMeasureCd": "MB",  "unitOfMeasureTxt": "MB",  "sharedUsageList": [  {  "phoneNumber": "4161514773",  "usedAmt": 100  }, {  "phoneNumber": "4161515165",  "usedAmt": 0  }  ]  }  **sharedUsageDetails**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | unitOfMeasureCd | string | Unit of measures for the usage elements within the sharedUsageList |  |  | | unitOfMeasureTxt | string | Localized description for the unitOfMeasureCd. |  |  | | sharedUsageList | List sharedUsage | List of shared usage elements. Each shared usage element in the represents data or message use by a subscriber participated in the shared meter. |  |  |   **sharedUsage**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | phoneNumber | string | Holds the usage information within the data or Message elements grouped by zones and meterGroups |  |  | | usedAmt | decimal | chargeAmts in each meterGroups aggregated to either data or message level in dollars. |  |  | |

# UsageDetail 1.0/2.0 REST Service

This service provides operations in support of displaying summary and detailed usage information, including a "usage dashboard" with graphical meters representing usage against an allowance. Changes added for v 2.0 are highlighted.

## voice-detail

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/voice-detail | |
| **Method** | GET |
| **Description** | detailed call records for a given bill cycle, plus summary info |
| **Input** | **ban** – account number  **sub** – subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system – default is wireless (initially only KB will be supported) | | cycleperiod | which bill cycle to return – current, previous, two back | Optional | Possible values are **0** (default), **-1** (previous bill cycle), **-2** (2 cycles back) | | periodthresholds | comma-separated list of numbers from 0-23 | Optional | If provided, the service will return rolled minute totals for each period of the day; e.g. if "12,18" is passed as a value, we will return the total min for midnight to noon (12:00), noon to 6 p.m. (18:00), and 6 p.m. to midnight | | totalsonly | "y", "n" | Optional | If is "y" (yes) and period-thresholds provided, do not populate callList (makes for more compact response) | | cachedetail | "y", "n" | Optional | If is "y" (yes), and cacherows not provided, do not populate callList, but cache it and return a cache token. | | cacherows | 10, 20 | Optional | if cachedetail = "y", populate callList with only first x rows from result set; if cacherows = all, full results are returned along with cache token. | | filter | "ld", "roam", "charges", "duration" | Optional | used to filter the list and return a matching subset  **ld** – all calls with long distance charges  **roam** – all roaming calls  **charges** – all calls with charges; if threshold provided then all calls with charges >= the threshold  **duration** – calls with minutes >= the threshold value (see below) | | threshold | 15, 2.5 | Optional\* | \*Mandatory if filter="duration". If filter = "duration", list will be filtered to show only calls lasting equal to or greater than the threshold number of minutes. For "charges", we will filter to show only calls incurring charges greater than or equal to the dollar amount | | sort | "a" (ascending), "d" (descending) | Optional | If filter is not set, sorts detail records chronologically ascending or descending (most recent first). Default as ascending. If filter is set, sorts as follows:  For filter=ld, roam or charges – sort by dollar amount; for filter=duration, sort by minutes duration. | | sortfield | "charges", "duration", "date" | Optional | overrides default filter sorting. "charges" – sort by totalAmt; "duration" – sort by durationAmt; "date" – sort by startDt | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 | IS | Invalid subscriber | Sub [905-333-5454] Not Found | exception.getMessage | UnknownSubscriberException | | 400 | IB | Invalid BAN | BAN xxxxx does not match sub yyyyyyyy |  | findSubscriber succeeds but account doesn't match provided BAN | | 400 |  | Invalid periodthresholds |  |  | numbers aren't 0-23 | | 400 |  | Invalid cycleperiod |  |  | must be 0, -1, or -2 | | 400 |  | Invalid cache parameters |  |  | cachedetail or cacherows invalid | | 500 |  | TelusAPIException |  | exception.getMessage | any general API exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “voiceUsage”: {  "cycleStartDt" : <date>,  "cycleEndDt" : <date>,  "callTotalNum" : <int>,  "callList" : [callDetail],  "callListCacheKey": <string>,  "callSummaryList" : [callSummary],  "callTotals"  },  "status" : <status>  where **callDetail** is composed of:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | startDt | date/time | Start of the call |  | | directionCd | string | indicates incoming "I" or outgoing "O" | "I", "O" | | roamingInd | boolean | indicates whether call is local or roaming | true, false | | durationAmt | double | minutes of call duration | 4.0 | | airtimeAmt | double | airtime charge | 0.0, 3.5 | | tollAmt | double | toll charge | 0.0, 2.5 | | additionalAmt | double | additional charge | 0.0 | | totalAmt | double | total charges | 4.5 | | toCityTxt | string | city called | TORONTO | | toStateTxt | string | state called | ON | | fromCityTxt | string | call origination city | MONTREAL | | fromStateTxt | string | call origination state | PQ | | locationTxt | string | description of location | BHON42142-49905-0850-0-1 | | phoneNumberTxt | string | number called |  | | rlhInd | boolean | Roam Like Home indicator | true, false – indicates whether the usage was roaming rated as domestic under the RLH scheme |   and **callSummary** is composed of:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | summaryDt | date | date of summary info |  | | callTotalNum | int | number of calls for the given day |  | | periodTotalsList | array of int | breakdown of minutes by period | 12.0, 2.0, 32.0 |   If **cachedetail** = "y", **callListCacheKey** will be populated. This will be used to retrieve call details from the cache.  and  **callTotals** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | totalDurationAmt | double | sum of durationAmt for all rows |  | | totalAirtimeAmt | double | sum of airtimeAmt for all rows |  | | totalTollAmt | double | sum of tollAmt for all rows |  | | totalAdditionalAmt | double | sum of additionalAmt for all rows |  | | grandTotalAmt | double | sum of totalAmt for all rows |  |   Example:  {  "voiceUsage": {  "cycleStartDt": "2013-08-02T00:00:00",  "cycleEndDt": "2013-08-02T00:00:00",  "callTotalNum": 2,  "callList": [  {  "startDt": "2013-08-07T07:24:23",  "directionCd": "I",  "roamingInd": false,  "durationAmt": 8.0,  "airtimeAmt": 5.0,  "tollAmt": 0.0,  "toCityTxt": "TORONTO",  "toStateTxt": "ON",  "fromCityTxt": "MONTREAL",  "fromStateTxt": "PQ",  "additionalAmt": 0.0,  "totalAmt": 0.0,  "locationTxt": "BHON42142-49905-0850-0-1, TORONTO, ON",  "phoneNumberTxt": "4165447099"  }, {  "startDt": "2013-08-09T09:27:05",  "directionCd": "I",  "roamingInd": false,  "durationAmt": 17.0,  "airtimeAmt": 12.0,  "tollAmt": 0.0,  "toCityTxt": "TORONTO",  "toStateTxt": "ON",  "fromCityTxt": "MONTREAL",  "fromStateTxt": "PQ",  "additionalAmt": 0.0,  "totalAmt": 0.0,  "locationTxt": "BHON40610-42290-0850-0-2, MISSISSAUGA, ON",  "phoneNumberTxt": "4165447099"  }  ],  "callSummaryList": [  {  "summaryDt": "2013-08-09T00:00:00",  "callTotalNum": 12,  "periodTotalsList": [  17.0, 5.0, 47.0  ]  }, {  "summaryDt": "2013-08-10T00:00:00",  "callTotalNum": 5,  "periodTotalsList": [  12.0, 2.0, 23.0  ]  }, {  "summaryDt": "2013-08-11T00:00:00",  "callTotalNum": 8,  "periodTotalsList": [  0.0, 15.0, 95.0  ]  }  ],  "callTotals" : {  "totalDurationAmt" : 27.0,  "totalAirtimeAmt" : "44.35,  "totalTollAmt" : "3.81",  "totalAdditionalAmt" : "0.0",  "grandTotalAmt" : "75.16"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **ClientAPI**  Account.getInvoiceHistory  Subscriber.getUnbilledCalls (for current cycle)  Subscriber.getBilledCalls (for previous cycle) |  |   Expected average response time: 3000 ms |
| **Sample GUI** |  |

## cached-call-detail

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/cached-call-detail | |
| **Method** | GET |
| **Description** | detailed call records for a given bill cycle, plus summary info |
| **Input** | **ban** – account number  **sub** – subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | ~~cacheId~~ cacheid | "3XSD-34099-2343-FXC" | Mandatory | Key used to retrieve cached call records | | filter | "ld", "roam", "charges", "duration" | Optional | used to filter the list and return a matching subset (see **voice-detail** for details) | | start | 10, 20, 30 | no | Starting row of (filtered) result set (1-based list) | | rows | 20, 40 | no | number of rows to return starting with "start" row | | threshold | 15, 2.5 | Optional\* | \*Mandatory if filter="duration" or "cost". If filter = "duration", list will be filtered to show only calls lasting equal to or greater than the threshold number of minutes. For "cost", we will filter to show only calls incurring charges greater than or equal to the dollar amount | | sort | "a" (ascending), "d" (descending) | Optional | If filter is not set, sorts detail records chronologically ascending or descending (most recent first). Default as ascending. If filter is set, sorts as follows:  For filter=ld, roam or charges – sort by dollar amount; for filter=duration, sort by minutes duration. | | sortfield | "charges", "duration", "date" | Optional | overrides default filter sorting. "charges" – sort by totalAmt; "duration" – sort by durationAmt; "date" – sort by startDt | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IC | Invalid cache |  |  | cache object not found or expired | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “cachedCallDetailResponse”: {  "callList" : [callDetail],  "status" : <status>  }  See **voice-detail** operation above for description of **callDetail**. |
| **SLA** | Services/APIs called:  None – retrieved from cache.  Expected average response time: 200 ms |
| **Sample GUI** |  |

# UsagePreferenceManagement REST Service

This service provides operations in support of usage threshold and notication delivery management – e.g. Setting a domestic or international allowance of 80% notification and pay per use blocking and notify at $50, and to be notified by email and SMS delivery preferences.

## subscriber-threshold-preferences

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/subscriber-threshold-preferences | |
| **Method** | GET |
| **Description** | Retrieve a subscriber’s threshold preferences (zones such as domestic, roaming and individual and shared thresholds) |
| **XSD reference** | Refer to element: getSubscriberThresholdPreference and getSubscriberThresholdPreferenceResponse |
| **Input** | **ban** – account number (postpaid)  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | subid |  | N | CRDB / wireless CODS subscription id | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid subscriber | Underlying error code | Subscriber not found: BAN [707020721], phone number [5141801193] | Subscriber not found | | 500 |  | WirelessCustomerPreferenceManagementService  getSubscriberNotificationThreshold call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “subscriberThresholdPreferenceResponse”:  {  “subscriberThresholdPreference”: {  “subscriptionId”: string,  “sub”: string,  “ban”: string,  “subscriberPreference”: { <SubscriberPreference> },  “subscriberThresholdList”: [ <SubscriberThreshold> ],  “status”: <status>  }  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionId | String | subscription id | “203343421” | | sub | String | Subscriber number | “4165551212” | | ban | String | Account number | “78645443” | | subscriberPreference | <SubscriberPreference> | Indicator if subscriber is authorized to unblock itself from a threshold block scenario |  | | subscriberThresholdList | Array of <SubscriberThreshold> | Contains thresholds based on allowance (shared, individual), threshold exempt code and threshold lists and state as well as PPU thresholds based on fixed or recurring |  |   where SubscriberPreference is described as  {  “allowPPUAndUnblockInd”: boolean,  “authorizedNumberList”: [ string ],  }  SubscriberPreference   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | allowPPUAndUnblockInd | boolean | Indicator if the subscriber is authorized to unblock itself from a threshold block scenario | true, false | | authorizedNumberList | Array of string | A list of subscribers authorized to unblock threshold based blocks  (reserved for future) | Returns empty list (reserved future) |   where SubscriberThreshold is described as  {  “serviceTypeCd”: string,  “zoneCd”: string  “allowanceList”: [ <Allowance> ],  “usageList”: [ <Usage> ]  }  SubscriberThreshold   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | serviceTypeCd | string | Service type code | “WISP” and potentially other data service types in the future | | zoneCd | string | Zone code | “DOMESTIC”,  “ROAMING” | | allowanceList | <Allowance> | Contains threshold value and threshold types such as notify, block and notify, and not set | Note that allowance can only notify but at 100% it can be blocknotify | | usageList | <Usage> | Contains threshold values and threshold types for pay per use and recurring |  |   where Allowance is described as:  “allowanceList”:  {  “allowanceTypeCd”: string,  “exemptionCd”: string,  “allowanceThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean  }  Allowance   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | allowanceTypeCd | string | Type of allowance thresholds | “SHARED”,  “INDIVIDUAL” | | exemptionCd | string | Threshold exempt code, a series of exempt codes based on factors such as service agreement that define whether or not user thresholds are overrided by Offer level exemption. These states can be used to grey our user thresholds etc. | “N” - Not exempted  “C” – Customer level exemption  “S” – Share group exemption  “M”- Market level exemption  “O” – Offer level exemption | | allowanceThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | Note that allowance can only notify but at 100% it can be blocknotify | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the allowanceTypeCd – individual/shared and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false |   where Threshold is described as:  “usageList”:  {  “thresholdValueNum”: integer,  “thresholdTypeCd”: string  }  Threshold   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | thresholdValueNum | integer | threshold value | 90 | | thresholdTypeCd | string | Threshold type such as notification, block and notify, no set | “NOTIFICATION” – notify only  “BLOCKNOTIFY” – block and notify  “NOTSET” – not set |   where Usage is described as:  {  “usageTypeCd”: string  “exemptionCd”: string  “usageThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean  }  Usage   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageTypeCd | string | Usage type for PPU such as fixed and or recurring i.e $50 and recurring at every $100 | “FIXED”,  “RECURRING” | | exemptedCd | string | Threshold exempt code, a series of exempt codes based on factors such as service agreement that define whether or not user thresholds are overrided by Offer level exemption. These states can be used to grey our user thresholds etc. | “N” - Not exempted  “C” – Customer level exemption  “S” – Share group exemption  “M”- Market level exemption  “O” – Offer” – not set | | usageThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | See Threshold definition above | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the usageTypeCd – fixed/recurring and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false |   Example  {"subscriberThresholdPreferenceResponse": {  "status": {  "statusCd": "200",  "statusTxt": "OK"  },  "ban": "70615002",  "sub": "6131950151",  "subscriberPreference": {"allowPPUAndUnblockInd": "true"},  "subscriberThresholdPreference": {"subscriberThresholdList": [  {  "serviceTypeCd": "WISP",  "zoneCd": "DOMESTIC",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTIFICATION"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTIFICATION"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTIFICATION"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "BLOCKNOTIFY"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "50",  "thresholdTypeCd": "BLOCKNOTIFY"  }],  "optOutNotificationInd": "false"  }  ]  },  {  "serviceTypeCd": "WISP",  "zoneCd": "ROAMING",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  }],  "optOutNotificationInd": "false"  }  ]  }  ]}  }} |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  getSubscriberNotificationThreshold | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** | **Thresholds and Opt out of notifications by zone, individual, shared**    **Allow PPU and Unblock** |

## subscriber-threshold-preferences (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/subscriber-threshold-preferences | |
| **Method** | PUT |
| **Description** | Update subscriber threshold preferences |
| **XSD reference** | Refer to element: updateSubscriberThresholdPreferenceList and updateSubscriberThresholdPreferenceListResponse |
| **Input** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Mandatory?** | **Possible/typical values** | | sub | string | Subscriber number | Y | “4165551212” | | ban | string | Account number | Y | “78645443” | | uuid | string | uuid | N\* | “e023490939fe”, not mandatory if contact role is device user | | contactRoleCd | string | Contact role performing the update | Y | “DEVICE”,  “OWNER”,  “MANAGER”,  “MEMBER” | | subscriptionId | string | KB subscription id | N\* | “203343421”, mandatory only for device and member contact roles | | subscriberThresholdList | Array of <SubscriberThreshold> | Contains thresholds based on allowance (shared, individual), threshold exempt code and threshold lists and state as well as PPU thresholds based on fixed or recurring | Y |  |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  “sub”: string,  “ban”: string,  “uuid”: string,  “contactRoleCd”: string,  “subscriptionId”: string,  “subscriberThresholdPreferenceList”: [ < SubscriberThresholdPreference > ]  “originatorInfo”: <OriginatorInfo>  }  where SubscriberThreshold is described as  {  “serviceTypeCd”: string,  “zoneCd”: string  “allowanceList”: [ <Allowance> ],  “usageList”: [ <Usage> ]  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Mandatory?** | **Possible/typical values** | | serviceTypeCd | string | Service type code | N | Default to “WISP” | | zoneCd | string | Zone code | Y | “DOMESTIC”,  “ROAMING” | | allowanceList | <Allowance> | Contains threshold value and threshold types such as notify, block and notify, and not set | Y | Note that allowance can only notify but at 100% it can be blocknotify | | usageList | <Usage> | Contains threshold values and threshold types for pay per use and recurring | Y |  |   where Allowance is described as:  {  “allowanceTypeCd”: string,  “allowanceThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean  }  Allowance (All fields Mandatory)   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | allowanceTypeCd | string | Type of allowance thresholds | “SHARED”,  “INDIVIDUAL” | | allowanceThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | Note that allowance can only notify but at 100% it can be blocknotify | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the allowanceTypeCd – individual/shared and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false |   where Threshold is described as:  {  “thresholdValueNum”: integer,  “thresholdTypeCd”: string  }  Threshold (All fields Mandatory)   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | thresholdValueNum | integer | threshold value | 90 | | thresholdTypeCd | string | Threshold type such as notification, block and notify, no set | “NOTIFICATION” – notify only  “BLOCKNOTIFY” – block and notify  “NOTSET” – not set |   where Usage is described as:  {  “usageTypeCd”: string  “usageThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean  }  Usage (All fields Mandatory)   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageTypeCd | string | Usage type for PPU such as fixed and or recurring i.e $50 and recurring at every $100 | “FIXED”,  “RECURRING” | | usageThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | See Threshold definition above | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the usageTypeCd – fixed/recurring and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false |   Where OriginatorInfo is described as  “originatorInfo”:  {  “userId”: “[telus@telus.com](mailto:telus@telus.com)”,  “applicationId”: “23”,  “userTypeCd”: “4”  }  Originator is not a means of authentication, only for provider logging purposes.  OriginatorInfo   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | userId | string | “[telus@telus.com](mailto:telus@telus.com)” | Y | Email address | | applicationId | string | 23 | Y | Application Id for beta is 23 | | userTypeCd | string | 4 | N | userTypeCd is always 4 for beta | | salesRepId | string |  | N |  | | channelOrgId | string |  | N |  | | outletId | string |  | N |  | | correlationId | string |  | N |  | | timestamp | string |  | N |  |   Example  {  "ban": "70615002",  "sub": "6131950151",  "contactRoleCd": "DEVICE",  "subscriberThresholdPreference": {"subscriberThresholdList": [  {  "serviceTypeCd": "WISP",  "zoneCd": "DOMESTIC",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTIFY"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTIFICATION"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTIFICATION"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "BLOCKNOTIFY"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "50",  "thresholdTypeCd": "BLOCKNOTIFY"  }],  "optOutNotificationInd": "false"  }  ]  },  {  "serviceTypeCd": "WISP",  "zoneCd": "ROAMING",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTIFY"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  }],  "optOutNotificationInd": "false"  }  ]  }  ]},  "originatorInfo": {  "userId": "simon.lee@telus.com",  "applicationId": "23",  "userTypeCd": "4"  }  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid subscriber | Underlying error code | Subscriber not found: BAN [707020721], phone number [5141801193] | Subscriber not found | | 500 |  | WirelessCustomerPreferenceManagementService  updateNotificationThreshold call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example  {  "status":  {  "statusCd": "200",  "statusTxt": "OK",  "systemErrorCd": 0,  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  updateNotificationThreshold | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** | **Thresholds and Opt out of notifications by zone, individual, shared** |

## subscriber-notification-preferences

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/subscriber-notification-preferences | |
| **Method** | GET |
| **Description** | Retrieve notification delivery preferences for a given ban, subscriber and all account owner and managers (when applicable) |
| **XSD reference** | Refer to element: getSubscriberNotificationPreferenceList and getSubscriberNotificationPreferenceListResponse |
| **Input** | Search by ban/sub will return a list of all subscriber notification preferences including all supporting roles under the ban. This is not mandatory because of the future search criterias listed below. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | WirelessCustomerPreferenceManagementService  getSubscriberNotificationAlertPreference call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “subscriberNotificationPreferenceListResponse”:  {  [ “subscriberNotificationPreferenceList”: <SubscriberNotificationPreferences> ],  “status”: <Status>  }  Where SubscriberNotificationPreferences can be described as:  {  “ban”: string,  “uuid”: string,  “subscriptionId”: string,  “contactRoleCd”: string,  “emailNotification”: { <EmailNotificationType> },  “smsNotification”: { <SMSNotificationType> },  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | uuid | string | profile uuid | “e0334234d333034”  And will be null for device roles that have no profile | | subscriptionId | string | KB subscription id | “203343421” | | ban | string | Account number | “78645443” | | contactRoleCd | string | contact role of the subscriber retrieved | “DEVICE” – device user (no profile)  “OWNER” – account owner  “MANAGER” – account manager  “MEMBER” – account member | | billingAccountTypeCd | string | BILLING\_ACCOUNT\_EXT\_TYP in refpds types to be read and passed back to provider | B - Business,C - Corporate, E - Exception, I - Consumer | | emailNotification | < EmailNotificationType > |  |  | | smsNotification | < SMSNotificationType > |  |  |   where EmailNotificationType is described as  {  “emailAddressTxt”: string,  “notifyInd”: boolean  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | emailAddressTxt | string | Notification email address | “telus@telus.com” | | notifyInd | boolean | Indicator for the delivery type whether to notify or not | true, false |   where SMSNotificationType is described as  {  “smsTxt”: string,  “notifyInd”: boolean  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | smsTxt | string | Notification sms | “4165551212” | | notifyInd | boolean | Indicator for the delivery type whether to notify or not | true, false |   Example  {  "subscriberNotificationPreferenceListResponse": {  "status": {  "statusCd": "200",  "statusTxt": "OK"  },  "subscriberNotificationPreferenceList": [  {  "uuid": "63f21181-3669-46ff-9589-f355e67cac9a",  "contactRoleCd": "OWNER",  "subscriptionId": "8322140",  "ban": "70638678",  "billingAccountTypeCd": "B",  "emailNotification": {  "emailAddressTxt": "usage1@ci-qa.com",  "notifyInd": "false"  },  "smsNotification": {  "smsTxt": "4160712656",  "notifyInd": "true"  }  },  {  "uuid": "63f21181-3669-46ff-9589-f355e67cac9a",  "contactRoleCd": "MEMBER",  "subscriptionId": "8322140",  "ban": "70638678",  "billingAccountTypeCd": "B",  "emailNotification": {  "emailAddressTxt": "usage1@ci-qa.com",  "notifyInd": "false"  },  "smsNotification": {  "smsTxt": "4160712656",  "notifyInd": "false"  }  }  ]  }} |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  getSubscriberNotificationThreshold | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## subscriber-notification-preferences (PUT)

|  |  |
| --- | --- |
| **OPERATION**  subscriber-notification-preferences | |
| **Method** | PUT |
| **Description** | Update subscriber preferences such as allow ppu and unblock indicator |
| **XSD reference** | Refer to element: updateSubscriberPreference |
| **Input** | {  “subscriberNotificationPreferenceList”: [ <SubscriberNotificationPreferences> ],  “originatorInfo”: <OriginatorInfo>  }  where SubscriberNotificationPreferences can be described as:  {  “ban”: string,  “billingAccountTypeCd”: string,  “subscriptionId”: string,  “contactRoleCd”: string,  “emailNotification”: { <EmailNotificationType> },  “smsNotification”: { <SMSNotificationType> }  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Mandatory?** | **Possible/typical values** | | ban | string | ban | Y | “70432323” | | subscriptionId | String | subscription ID | N\* | “20452012”, mandatory only for device and member contact roles | | uuid | String | uuid | N\* | “e02ef3023120”, mandatory only if contact role is owner, manager, member (device user has no uuid) | | billingAccountTypeCd | string | BILLING\_ACCOUNT\_EXT\_TYP in refpds types to be read and passed back to provider | Y | B - Business,C - Corporate, E - Exception, I - Consumer | | contactRoleCd | string | contact role of the subscriber performing the update | Y | “DEVICE” – device user (no profile)  “OWNER” – account owner  “MANAGER” – account manager  “MEMBER” – account member | | emailNotification | < EmailNotificationType > |  | N | emailNotification | | smsNotification | < SMSNotificationType > |  | N | smsNotification |   where EmailNotificationType is described as  {  “emailAddressTxt”: string,  “notifyInd”: boolean  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | emailAddressTxt | string | Notification email address | “telus@telus.com” | | notifyInd | boolean | Indicator for the delivery type whether to notify or not | true, false |   where SMSNotificationType is described as  {  “smsTxt”: string,  “notifyInd”: boolean  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | smsTxt | string | Notification sms | “4165551212” | | notifyInd | boolean | Indicator for the delivery type whether to notify or not | true, false |   Where OriginatorInfo is described as  "originatorInfo": {  "userId": "telus@telus.com",  "userTypeCd": "4",  "applicationId": "23",  }    Originator is not a means of authentication, only for provider logging purposes.  OriginatorInfo   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | userId | string | “[telus@telus.com](mailto:telus@telus.com)” | Y | Email address | | applicationId | string | 23 | Y | Application Id for beta is 23 | | userTypeCd | string | 4 | N | userTypeCd is always 4 for beta | | salesRepId | string |  | N |  | | channelOrgId | string |  | N |  | | outletId | string |  | N |  | | correlationId | string |  | N |  | | timestamp | string |  | N |  |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {    "subscriberNotificationPreferenceList": [  {  "uuid": "63f21181-3669-46ff-9589-f355e67cac9a",  "contactRoleCd": "OWNER",  "subscriptionId": "8322140",  "ban": "70638678",  "billingAccountTypeCd": "B",  "emailNotification": {  "emailAddressTxt": "usage1@ci-qa.com",  "notifyInd": "true"  },  "smsNotification": {  "smsTxt": "4165551212",  "notifyInd": "true"  }  },  {  "uuid": "63f21181-3669-46ff-9589-f355e67cac9a",  "contactRoleCd": "MEMBER",  "subscriptionId": "8322140",  "ban": "70638678",  "billingAccountTypeCd": "B",  "emailNotification": {  "emailAddressTxt": "usage1@ci-qa.com",  "notifyInd": "false"  },  "smsNotification": {  "smsTxt": "4160712656",  "notifyInd": "false"  }  }  ],  "originatorInfo": {  "userId": "telus@telus.com",  "userTypeCd": "4",  "applicationId": "23",  }  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | WirelessCustomerPreferenceManagementService  updateSubscriberNotificationAlertPreference call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example  {  "status":  {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  updateSubscriberNotificationAlertPreference | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## subscriber-payperuse-preferences (PUT)

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/sub/{sub}/subscriber-payperuse-preferences | |
| **Method** | PUT |
| **Description** | Update subscriber pay per use preference such as allow PPU and unblock |
| **XSD reference** | Refer to element: updateSubscriberPayperusePreference and updateSubscriberPayperusePreferenceResponse |
| **Input** | {  “ban”: string,  “sub”: string,  “uuid”: string,  “contactRoleCd”: string,  “subscriberPreference”: {  “allowPPUAndUnblockInd”: boolean,  “authorizedNumberList”: [ string ]  },  “originatorInfo”: < OriginatorInfo >  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Mandatory?** | **Possible/typical values** | | ban | string | ban number | Y | “70432323” | | sub | string | subscriber number | Y | “4165551212” | | uuid | string | uuid | N\* | “e02ef3023120”, mandatory only if contact role is owner, manager, member | | contactRoleCd | string | contact role of the subscriber retrieved | Y | “DEVICE” – device user (no profile)  “OWNER” – account owner  “MANAGER” – account manager  “MEMBER” – account member | | subscriberPreference | <SubscriberPreferenceType> |  | Y |  |   where SubscriberPreferenceType is described as  {  “allowPPUAndUnblockInd”: boolean,  “authorizedNumberList”: [ string ]  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Mandatory?** | **Possible/typical values** | | allowPPUAndUnblockInd | boolean | Indicator if the subscriber is authorized to unblock itself from a threshold block scenario | Y | true, false | | authorizedNumberList | Array of string | A list of subscribers authorized to unblock threshold based blocks  (reserved for future) | N | Returns empty list (reserved future) |   Where OriginatorInfo is described as  "originatorInfo": {  "userId": "telus@telus.com",  "userTypeCd": "4",  "applicationId": "23",  }  Originator is not a means of authentication, only for provider logging purposes.  OriginatorInfo   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | userId | string | “[telus@telus.com](mailto:telus@telus.com)” | Y | Email address | | applicationId | string | 23 | Y | Application Id for beta is 23 | | userTypeCd | string | 4 | N | userTypeCd is always 4 for beta | | salesRepId | string |  | N |  | | channelOrgId | string |  | N |  | | outletId | string |  | N |  | | correlationId | string |  | N |  | | timestamp | string |  | N |  |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  “ban”: “70432323”,  “sub”: “4165551212”,  “uuid”: “e02ef3023120”,  “contactRoleCd”: “MEMBER”,  “subscriberPreference”: {  “allowPPUAndUnblockInd”: true  },  "originatorInfo": {  "userId": "telus@telus.com",  "userTypeCd": "4",  "applicationId": "23",  }  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | WirelessCustomerPreferenceManagementService  updateSubscriberPreference call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example  {  "status":  {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  updateSubscriberPreference | 2000 ms average; 5000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## validate-sms-number

|  |  |
| --- | --- |
| **OPERATION**  /sub/{sub}/validate-sms-number | |
| **Method** | GET |
| **Description** | Validate an SMS notification preference number by NetworkSubscriberProfileService\_V4 |
| **XSD reference** | Refer to element: validateSMSPreferenceNumber and validateSMSPreferenceNumberResponse |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | |  |  |  |  | |  |  |  |  | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid Telus or Koodo subscriber number or is in cancelled state | Subscriber not found or is in cancelled state for phone number: 4165765688 | Underlying error code | Subscriber not found in Telus or Koodo network or is in cancelled state | | 500 |  | NetworkSubscriberProfileService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example  {  "status":  {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **NetworkSubscriberProfileService\_V4\_0** getWirelessNetworkSubscriberProfile | 650 ms average; 1300 ms peak |   Expected average response time: 950 ms |
| **Sample GUI** |  |

# UsagePreferenceManagement 1.1 REST Service

This service extends from version 1.0 (see above). Below are the updated operations.

## subscriber-threshold-preferences

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/subscriber-threshold-preferences | |
| **Method** | GET |
| **Description** | Retrieve a subscriber’s threshold preferences (zones such as domestic, roaming and individual and shared thresholds) |
| **XSD reference** | Refer to element: getSubscriberThresholdPreference and getSubscriberThresholdPreferenceResponse |
| **Input** | **ban** – account number (postpaid)  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | subid |  | N | CRDB / wireless CODS subscription id | | allowancecheck | y, n | N | flag to request a check on what allowances are present on the subscriber profile.  If “allowancecheck “ was passed as ‘y’, the service will perform the allowances check for the given subscriber and each Allowance element in the response will have “allowanceOnProfileInd” field set appropriately.  If allowancecheck is passed as ‘n’ or not passed at all, no allowances check will be performed and “allowanceOnProfileInd” will not be returned. | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid subscriber | Underlying error code | Subscriber not found: BAN [707020721], phone number [5141801193] | Subscriber not found | | 500 |  | WirelessCustomerPreferenceManagementService  getSubscriberNotificationThreshold call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | “subscriberThresholdPreferenceResponse”:  {  “subscriberThresholdPreference”: {  “subscriptionId”: string,  “sub”: string,  “ban”: string,  “subscriberPreference”: { <SubscriberPreference> },  “subscriberThresholdList”: [ <SubscriberThreshold> ],  “status”: <status>  }  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionId | String | subscription id | “203343421” | | sub | String | Subscriber number | “4165551212” | | ban | String | Account number | “78645443” | | subscriberPreference | <SubscriberPreference> | Indicator if subscriber is authorized to unblock itself from a threshold block scenario |  | | subscriberThresholdList | Array of <SubscriberThreshold> | Contains thresholds based on allowance (shared, individual), threshold exempt code and threshold lists and state as well as PPU thresholds based on fixed or recurring |  |   where SubscriberPreference is described as  {  “allowPPUAndUnblockInd”: boolean,  “authorizedNumberList”: [ string ],  }  SubscriberPreference   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | allowPPUAndUnblockInd | boolean | Indicator if the subscriber is authorized to unblock itself from a threshold block scenario | true, false | | authorizedNumberList | Array of string | A list of subscribers authorized to unblock threshold based blocks  (reserved for future) | Returns empty list (reserved future) |   where SubscriberThreshold is described as  {  “serviceTypeCd”: string,  “zoneCd”: string  “allowanceList”: [ <Allowance> ],  “usageList”: [ <Usage> ]  }  SubscriberThreshold   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | serviceTypeCd | string | Service type code | “WISP” and potentially other data service types in the future | | zoneCd | string | Zone code | “DOMESTIC”,  “ROAMING” | | allowanceList | <Allowance> | Contains threshold value and threshold types such as notify, block and notify, and not set | Note that allowance can only notify but at 100% it can be blocknotify | | usageList | <Usage> | Contains threshold values and threshold types for pay per use and recurring |  |   where Allowance is described as:  “allowanceList”:  {  “allowanceTypeCd”: string,  “exemptionCd”: string,  “allowanceThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean,  “allowanceOnProfileInd”: boolean  }  Allowance   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | allowanceTypeCd | string | Type of allowance thresholds | “SHARED”,  “INDIVIDUAL” | | exemptionCd | string | Threshold exempt code, a series of exempt codes based on factors such as service agreement that define whether or not user thresholds are overrided by Offer level exemption. These states can be used to grey our user thresholds etc. | “N” - Not exempted  “C” – Customer level exemption  “S” – Share group exemption  “M”- Market level exemption  “O” – Offer level exemption | | allowanceThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | Note that allowance can only notify but at 100% it can be blocknotify | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the allowanceTypeCd – individual/shared and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false | | allowanceOnProfileInd |  | indicator denoting if the subscriber has the allowance of this type (INDIVIDUAL, SHARED) on his profile. Used for retrieve operation only. | true,  false |   where Threshold is described as:  “usageList”:  {  “thresholdValueNum”: integer,  “thresholdTypeCd”: string  }  Threshold   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | thresholdValueNum | integer | threshold value | 90 | | thresholdTypeCd | string | Threshold type such as notification, block and notify, no set | “NOTIFICATION” – notify only  “BLOCKNOTIFY” – block and notify  “NOTSET” – not set |   where Usage is described as:  {  “usageTypeCd”: string  “exemptionCd”: string  “usageThresholdList”: [ <Threshold> ],  “optOutNotificationInd”: boolean  }  Usage   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageTypeCd | string | Usage type for PPU such as fixed and or recurring i.e $50 and recurring at every $100 | “FIXED”,  “RECURRING” | | exemptedCd | string | Threshold exempt code, a series of exempt codes based on factors such as service agreement that define whether or not user thresholds are overrided by Offer level exemption. These states can be used to grey our user thresholds etc. | “N” - Not exempted  “C” – Customer level exemption  “S” – Share group exemption  “M”- Market level exemption  “O” – Offer” – not set | | usageThresholdList | Array of <Threshold> | Contains threshold value and threshold types such as notify, block and notify, and not set | See Threshold definition above | | optOutNotificationInd | boolean | Opt out of threshold notification indicator – this is the master preference setting that can turn off notifications based on the usageTypeCd – fixed/recurring and the zoneCd – domestic/roaming  Note the threshold settings are still retained, but acts as a master on off switch | true,  false |   Example  {"subscriberThresholdPreferenceResponse": {  "status": {  "statusCd": "200",  "statusTxt": "OK"  },  "ban": "70615002",  "sub": "6131950151",  "subscriberPreference": {"allowPPUAndUnblockInd": "true"},  "subscriberThresholdPreference": {"subscriberThresholdList": [  {  "serviceTypeCd": "WISP",  "zoneCd": "DOMESTIC",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceOnProfileInd": "true",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTIFICATION"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTIFICATION"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTIFICATION"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "BLOCKNOTIFY"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "50",  "thresholdTypeCd": "BLOCKNOTIFY"  }],  "optOutNotificationInd": "false"  }  ]  },  {  "serviceTypeCd": "WISP",  "zoneCd": "ROAMING",  "allowanceList": [ {  "allowanceTypeCd": "INDIVIDUAL",  "exemptionCd": "N",  "allowanceOnProfileInd": "false",  "allowanceThresholdList": [  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "80",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "95",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  }],  "usageList": [  {  "usageTypeCd": "FIXED",  "exemptedCd": "N",  "usageThresholdList": [  {  "thresholdValueNum": "50",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "25",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "10",  "thresholdTypeCd": "NOTSET"  },  {  "thresholdValueNum": "75",  "thresholdTypeCd": "NOTSET"  }  ],  "optOutNotificationInd": "false"  },  {  "usageTypeCd": "RECURRING",  "exemptedCd": "N",  "usageThresholdList": [ {  "thresholdValueNum": "100",  "thresholdTypeCd": "NOTSET"  }],  "optOutNotificationInd": "false"  }  ]  }  ]}  }} |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v1\_0**  getSubscriberNotificationThreshold | 2000 ms average; 5000 ms peak | | **CustomerRatingInquiryService\_v5\_0**  getDetailUsageMeterList | 1000 ms average; 3000 ms peak |   Expected average response time: 2500 – 3000 ms |
| **Sample GUI** | **Thresholds and Opt out of notifications by zone, individual, shared**    **Allow PPU and Unblock** |

# UsageBlockManagement REST Service

This service provides operations in support of usage data block management – e.g. retrieve data block status, unblock threshold etc.

## data-block-status

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/data-block-status | |
| **Method** | GET |
| **Description** | Retrieve data block statuses |
| **XSD reference** | Refer to element: getDataBlockStatusList and getDataBlockStatusListResponse |
| **Input** | **ban** – account number (postpaid)  **sub** – phone number |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid phone number for BAN - Request is invalid |  |  |  | | 500 |  | OnlineChargingSubscriberAccountService call failed | Underlying error code | Underlying error message | Any other Policy or Service Exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “dataBlockStatusListReponse”:  {  “dataBlockStatusList”: [ <DataBlockStatus> ]  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | dataBlockStatusList | Array of <DataBlockStatus> | Data block status |  |   where DataBlockStatus is described as:  [  {  “zoneCd”: string,  “actionCd”: string,  “dataBlockTypeCd”: string  “blockAmt”: double  }  ]  DataBlockStatus   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | zoneCd | string | Domestic, Roaming | DOMESTIC, ROAMING | | actionCd | string | Data block or unblock action | UNBLOCKED,  BLOCKED\_PPU (OPTIONAL and PERMANENT only),  BLOCKED\_ALL (immediate block only) | | dataBlockTypeCd | string | Data block types | OPTIONAL,  PERMANENT,  IMMEDIATE (for future) | | blockAmt | double | Threshold blocked amount | 50.00 |   Example  {  "dataBlockStatusList": {"dataBlockStatus": [  {  "zoneCd": "DOMESTIC",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "PERMANENT",  "blockAmt": "0.0"  },  {  "zoneCd": "DOMESTIC",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "IMMEDIATE",  "blockAmt": "0.0"  },  {  "zoneCd": "DOMESTIC",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "OPTIONAL",  "blockAmt": "0.0"  },  {  "zoneCd": "ROAMING",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "PERMANENT",  "blockAmt": "0.0"  },  {  "zoneCd": "ROAMING",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "IMMEDIATE",  "blockAmt": "0.0"  },  {  "zoneCd": "ROAMING",  "actionCd": "UNBLOCKED",  "dataBlockTypeCd": "OPTIONAL",  "blockAmt": "0.0"  }  ]},  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }  Additional Notes:  \*\* Update: Our provider does not return blocked status under allowances  i.e. 100% allowance blockAndNotify  -Permanent and Optional when blocked return BLOCKED\_PPU only (as mentioned above, allowance block statuses cannot be determined at this time).  -Immediate Blocks only return BLOCKED\_ALL.  -blockAmt is only applicable for OPTIONAL and PERMANENT blocked statuses.  -Currently SAPCC rating engine does not support CDMA roaming block and unblock states. |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **OnlineChargingSubscriberAccountService\_v2\_0**  getSubscriberBlockStatus | 5000 ms average; 7000 ms peak |   Expected average response time: 5300 ms |
| **Sample GUI** |  |

## data-block-status (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/data-block-status | |
| **Method** | PUT |
| **Description** | Update a block status i.e. unblock a threshold block |
| **XSD reference** | Refer to element: setDataBlockStatus and setDataBlockStatusResponse |
| **Input** | Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | ban | string | ban number | Y |  | | sub | string | subscriber number | Y |  | | zoneCd | string | DOMESTIC,  ROAMING | Y | Zone information | | networkTypeCd | string | CDMA, HSPA (also covers hspa/lte) | Y | Network type | | dataEventTypeInfo | <DataEventType> |  | Y | See DataEventType | | originatorInfo | <OriginatorInfo> |  | Y | See OriginatorInfo |   Where DataEventType is described as  “dataEventTypeInfo”:  {  “actionCd”: “UNBLOCK”,  “dataBlockTypeCd”: “OPTIONAL”  }  DataEventType   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | actionCd | string | BLOCK, UNBLOCK | Y | Block or unblock action | | dataBlockTypeCd | string | OPTIONAL,  IMMEDIATE\* (reserved for future HSPA only),  Supported Self serve keystone update scenarios:  \*unblock threshold  \*block and unblock immediate (Please FR 687203 for details) | Y | The type of blocking mechanism |   where originatorInfo is described as  “originatorInfo”:  {  “userId”: “[telus@telus.com](mailto:telus@telus.com)”,  “applicationId”: “23”,  “userTypeCd”: “4”  }  Originator is not a means of authentication, only for provider logging purposes.  OriginatorInfo   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Datatype** | **Values** | **Mandatory?** | **Comments** | | userId | string | “[telus@telus.com](mailto:telus@telus.com)” | Y | Email address | | applicationId | string | 23 | Y | Application Id for beta is 23 | | userTypeCd | string | 4 | N | userTypeCd is always 4 for beta | | salesRepId | string |  | N |  | | channelOrgId | string |  | N |  | | outletId | string |  | N |  | | correlationId | string |  | N |  | | timestamp | string |  | N |  |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "ban": "70702072",  "sub": "5141801193",  "zoneCd": "DOMESTIC",  "networkTypeCd": "HSPA",  "dataEventTypeInfo": {  "actionCd": "UNBLOCK",  "dataBlockTypeCd": "OPTIONAL",  },  "originatorInfo":  {  "userId": "telus@telus.com",  "userTypeCd": "4",  "applicationId": "23"  }  }  Additional notes:  -IMMEDIATE blocks (IMMEDIATE should not be used until all Telus Koodo applications have been aligned to use IMMEDIATE as a blocking mechanism, this is for future use and for HSPA only).  -Currently SAPCC rating engine does not support CDMA roaming block and unblock states. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid phone number for BAN - Request is invalid |  |  |  | | 500 |  | WirelessDataToggleService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example  {  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessDataToggleService\_V2\_0**  toggleData | 1000 ms average; 3000 ms peak |   Expected average response time: 1300 ms |
| **Sample GUI** |  |

# UsageNotificationAcceptanceHistory REST Service

This service provides operations in support of usage notification, threshold and block and unblock consent history – e.g. retrieve notification history, notification delivery change history, block and unblock event and consent history.

## threshold-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/threshold-history | |
| **Method** | GET |
| **Description** | Retrieve threshold change history for a given ban, subscriber |
| **XSD reference** | Refer to element: getUsageThresholdChangeHistory and getUsageThresholdChangeHistoryResponse |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | DG\_SIS\_404 - Account number is invali. |  |  |  | | 400 |  | Invalid phone number passed. Expected format is 10 digits with no hyphens, spaces, brackets or international dialing codes. |  |  |  | | 400 |  | from date may not be null |  |  |  | | 400 |  | to date may not be null |  |  |  | | 400 |  | Invalid phone number for BAN - Request is invalid |  |  |  | | 500 |  | WirelessCustomerPreferenceManagementService call failed/returned an unexpected response | Underlying error code | Underlying error message | Any Policy or Service Exception | | 500 |  | BillingAccountDataGridService call failed | Underlying error code | Underlying error message | Any Policy or Service Exception | | 500 |  | general error  system and policy exceptions | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “usageThresholdChangeHistoryResponse”:  {  “usageThresholdChangeHistoryList”: [ < UsageThresholdChangeHistory > ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageThresholdChangeHistoryList | Array of <UsageThresholdChangeHistory> | Usage threshold change history |  | | status | <status> | status |  |   Where UsageThresholdChangeHistory is described as   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | originatorUuid | string | Originator uuid who performed the action | “e023490939fe” when applicable | | originatorContactRoleCd | string | Originator Contact role  RefPDS: CONTACT\_ROLE | DVC - device,  ACCTMBR - account member,  ACCTMGR - account manager,  ACCTOWNR – account owner | | usageThresholdUnitTypeCd | string | ThresholdUnitType  RefPDS: USAGE\_THRESHOLD\_UNIT\_TYPE | PERCENTAGE, DOLLAR,  NA | | usageThresholdTypeCd | string | ThresholdType  RefPDS: USAGE\_THRESHOLD\_TYPE | ALLOWANCE, OVERAGE, RECURRING,  NA | | networkCoverageAreaTypeCd | string | Network coverage area  RefPDS: NTWK\_CVRG\_AREA\_TYPE | DOMESTIC, ROAMING | | thresholdValue | long | Threshold value | 60, 70 etc | | usagePlanLevelTypeCd | string | UsagePlanLevelTypeCode  RefPDS:  USER\_PLAN\_LEVEL\_TYPE and remapped to values mentioned here | SHARED,  INDIVIDUAL | | usageNotifyControlCd | string | Block indicator for threshold  RefPDS:  USAGE\_NOTIF\_CTRL\_SET and remapped to values mentioned here  NA is a condition where it indicates that the threshold setting is not applicable  i.e. usageThresholdHistoryRecordTypeCd is OPTOUT and when it is THRESHOLD, you will see one of the three states possible for a threshold , NOTSET, NOTIFICATION, BLOCKNOTIFY | NOTIFICATION,  BLOCKNOTIFY,  NOTSET,  NA | | optOutAllNotificationAndBlockInd | boolean | Opt out from all Notifications and Blocks indicator | true, false or not present  (not present indicates this is not an opt out record) | | usageThresholdHistoryRecordTypeCd | string | Identifies the overall record type whether it is a threshold record or an optout record | THRESHOLD,  OPTOUT | | effectiveStartDt | date | Effective start date | e.g. “2013-06-07T08:50:44” | | effectiveEndDt | date | Effective end date | e.g. “2013-06-08T08:50:44”or a future date if threshold is still current | | originatorApplicationInfo | <OriginatorApplicationInfo> | Originating application information |  |   {  “originatorUuid”: string,  “originatorContactRoleCd”: string,  “usageThresholdUnitTypeCd”: string,  “usageThresholdTypeCd”: string,  “networkCoverageAreaTypeCd”: string,  “thresholdValue”: long,  “usagePlanLevelTypeCd” : string,  “usageNotifyControlCd”: string,  “optOutAllNotificationAndBlockInd” : boolean  “usageThresholdHistoryRecordCd” : string  “effectiveStartDate”: date,  “effectiveEndDt”: date,  “originatorApplicationInfo”: <OriginatorApplicationInfo>  }  Where OriginatorApplicationInfo is described as  OriginatorApplicationInfo   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Datatype** | **Description** | **Possible/typical values** | | userId | string | User or agent |  | | applicationId | string | Application id |  | | salesRepId | string | Sales rep id |  | | channelOrganizationId | string | Channel org id |  | | outletId | string | Outlet id |  |   Where Status is described as:  {  "statusCd": "200",  "statusTxt": "OK",  "systemErrorCd": 0,  }  **Extra notes:**  **Our provider in order to reduce records in database will no longer return the default records and only subsequent changed records.**  **i.e. If a subscriber is new and settings have never been changed, and you change a threshold at 70% from NOTSET to NOTIFICATION then you would not see an expired record of the 70% NOTSET but you would get the new record for 70% NOTIFICATION. Any subsequent changes after that you would see the expiring records. If you decided to change 70% NOTIFICATION to 70% NOTSET then you would see two records, one for 70% NOTIFICATION end date would be the date you made the change and the other record is 70% NOTSET and the end date would be a “current record” which is a future date.**  **Keep in mind this is a threshold record only as in the case above OptOut will be null or not present.**  **OptOut is in another record where it would return true or false and the threshold attributes would all be NA.** |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v2\_0**  retrieveUsageThresholdPreferencesChangeHistory | 2000 ms average; 5000 ms peak |   Expected average response time: 2300 ms |
| **Example** | Example of response:  {"usageThresholdChangeHistoryResponse": {  "usageThresholdChangeHistoryList": [  {  "usageThresholdUnitTypeCd": "PERCENTAGE",  "usageThresholdTypeCd": "ALLOWANCE",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "100",  "usagePlanLevelTypeCd": "SHARED",  "usageNotifyControlCd": "NOTSET",  "usageThresholdHistoryRecordTypeCd": "THRESHOLD",  "effectiveStartDt": "2014-11-21-05:00",  "effectiveEndDt": "9999-12-31-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "DOLLAR",  "usageThresholdTypeCd": "OVERAGE",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "50",  "usagePlanLevelTypeCd": "INDIVIDUAL",  "usageNotifyControlCd": "BLOCKNOTIFY",  "usageThresholdHistoryRecordTypeCd": "THRESHOLD",  "effectiveStartDt": "2014-11-21-05:00",  "effectiveEndDt": "2014-11-22-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "DOLLAR",  "usageThresholdTypeCd": "OVERAGE",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "100",  "usagePlanLevelTypeCd": "INDIVIDUAL",  "usageNotifyControlCd": "NOTIFICATION",  "usageThresholdHistoryRecordTypeCd": "THRESHOLD",  "effectiveStartDt": "2014-11-22-05:00",  "effectiveEndDt": "9999-12-31-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "DOLLAR",  "usageThresholdTypeCd": "OVERAGE",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "50",  "usagePlanLevelTypeCd": "INDIVIDUAL",  "usageNotifyControlCd": "NOTSET",  "usageThresholdHistoryRecordTypeCd": "THRESHOLD",  "effectiveStartDt": "2014-11-22-05:00",  "effectiveEndDt": "2014-11-25-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "DOLLAR",  "usageThresholdTypeCd": "OVERAGE",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "50",  "usagePlanLevelTypeCd": "INDIVIDUAL",  "usageNotifyControlCd": "BLOCKNOTIFY",  "usageThresholdHistoryRecordTypeCd": "THRESHOLD",  "effectiveStartDt": "2014-11-25-05:00",  "effectiveEndDt": "9999-12-31-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "NA",  "usageThresholdTypeCd": "NA",  "networkCoverageAreaTypeCd": "DOMESTIC",  "thresholdValue": "0",  "usagePlanLevelTypeCd": "SHARED",  "usageNotifyControlCd": "NA",  "optOutAllNotificationAndBlockInd": "true",  "usageThresholdHistoryRecordTypeCd": "OPTOUT",  "effectiveStartDt": "2014-11-25-05:00",  "effectiveEndDt": "2014-11-25-05:00",  "originatorApplicationInfo": {  "userId": "x108557",  "applicationId": "27"  }  },  {  "usageThresholdUnitTypeCd": "NA",  "usageThresholdTypeCd": "NA",  "networkCoverageAreaTypeCd": "ROAMING",  "thresholdValue": "0",  "usagePlanLevelTypeCd": "INDIVIDUAL",  "usageNotifyControlCd": "NA",  "optOutAllNotificationAndBlockInd": "false",  "usageThresholdHistoryRecordTypeCd": "OPTOUT",  "effectiveStartDt": "2014-11-25-05:00",  "effectiveEndDt": "9999-12-31-05:00",  "originatorApplicationInfo": {  "userId": "T837039",  "applicationId": "27"  }  }  ],  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }} |

## notification-contact-change-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/notification-contact-change-history | |
| **Method** | GET |
| **Description** | Retrieve notification contact change history such as changes to SMS contact number or email address changes |
| **XSD reference** | Refer to element: getNotificationContactChangeHistory and getNotificationContactChangeHistoryResponse |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | sub | “4165551212” | N | Pass in subscriber to refine search | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | BAN may not be null. |  |  | BAN parameter not provided | | 400 | IB | Account Number passed fails sanity check for this application. Please check if your account number is a valid Wireless BAN. |  |  |  | | 400 | IS | Invalid phone number passed. Expected format is ten digits with no hyphens, spaces, brackets or international dialing codes. |  |  |  | | 400 | IS | sub may not be null |  |  |  | | 400 | ID | from date may not be null |  |  |  | | 400 | ID | to date may not be null |  |  |  | | 400 | IB | Invalid BAN and/or Phone Number |  |  | SIS Domain service validation failure. | | 500 |  | WirelessCustomerPreferenceManagementService call failed/returned an unexpected response | Underlying error code | Underlying error message | Any Policy or Service Exception | | 500 |  | BillingAccountDataGridService call failed | Underlying error code | Underlying error message | Any Policy or Service Exception | | 500 |  | general error  system and policy exceptions | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “notificationContactChangeHistoryResponse”:  {  “notificationContactChangeHistoryList”: [ <NotificationContactChangeHistory> ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | notificationContactChangeHistoryList | Array of <NotificationContactChangeHistory> | Notification contact medium change history |  | | status | <status> | status |  |   Where NotificationContactChangeHistory is described as   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionId | string | Subscription id | “203343421” when applicable | | originatorUuid | string | Originator uuid who performed the action | “e023490939fe” when applicable | | originatorContactRoleCd | string | Originator Contact role  RefPDS: CONTACT\_ROLE | DVC - device,  ACCTMBR - account member,  ACCTMGR - account manager,  ACCTOWNR – account owner | | contactMechanismTypeCd | string | Contact mechanism type  RefPDS: CNTCT\_MECHANISM\_TYPE | SMS, EMAIL | | contactMechanismTxt | string | Contact mechanism type value | SMS number or email address | | effectiveStartDt | date | Effective start date | e.g. “2013-06-07T08:50:44” | | effectiveEndDt | date | Effective end date | e.g. “2013-06-08T08:50:44”or a future date if contact is still current | | originatorApplicationInfo | <OriginatorApplicationInfo> | Originating application information |  |   {  “subscriptionId”: string,  “originatorUuid”: string,  “originatorContactRoleCd”: string,  “contactMechanismTypeCd”: string,  “contactMechanismValueTxt”: string,  “effectiveStartDt”: date,  “effectiveEndDt”: date,  “originatorApplicationInfo”: <OriginatorApplicationInfo>  }  Where OriginatorApplicationInfo is described as  OriginatorApplicationInfo   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Datatype** | **Description** | **Possible/typical values** | | userId | string | User or agent |  | | applicationId | string | Application id |  | | salesRepId | string | Sales rep id |  | | channelOrgId | string | Channel org id |  | | outletId | string | Outlet id |  |   Where Status is described as:  {  "statusCd": "200",  "statusTxt": "OK",  "systemErrorCd": 0,  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v2\_0**  retrieveContactMediumPreferencesChangeHistory | 2000 ms average; 5000 ms peak |   Expected average response time: 2300 ms |

## notification-preference-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/notification-preference-history | |
| **Method** | GET |
| **Description** | Retrieve notification preference change history such as changes to opt in or out of SMS and or email |
| **XSD reference** | Refer to element: getNotificationPreferenceChangeHistory and getNotificationPreferenceChangeHistoryResponse |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | sub | “4165551212” | N | Pass in subscriber to refine search | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | WirelessCustomerPreferenceManagementService call failed | Underlying error code | Underlying error message | Any other Policy or Service Exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “notificationPreferenceChangeHistoryResponse”:  {  “notificationPreferenceChangeHistoryList”: [ < NotificationPrefChangeHistory > ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | notificationPrefChangeHistoryList | Array of < NotificationPreferenceChangeHistory > | Notification preference change history |  | | status | <status> | status |  |   Where NotificationPreferenceChangeHistory is described as  NotificationPreferenceChangeHistory   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionId | string | Subscription id | “203343421” when applicable | | originatorUuid | string | Originator uuid who performed the action | “e023490939fe” | | originatorContactRoleCd | string | Originator Contact role  RefPDS: CONTACT\_ROLE | DVC - device,  ACCTMBR - account member,  ACCTMGR - account manager,  ACCTOWNR – account owner | | contactMechanismTypeCd | string | Contact mechanism type  RefPDS: CNTCT\_MECHANISM\_TYPE | SMS, EMAIL | | contactOptinTypeCd | string | Contact opt in type  RefPDS: CNTCT\_OPTN\_TYPE | OPTIN – opt in for notification,  OPTOUT – opt out | | billingAccountTypeCd | string | Billing account type  RefPDS: BILLING\_ACCOUNT\_EXT\_TYP | B – business,  C – corporate,  E – exception,  I - consumer | | effectiveStartDt | date | Effective start date | e.g. “2013-06-07T08:50:44” | | effectiveEndDt | date | Effective end date | e.g. “2013-06-08T08:50:44”or a future date if preference is still current | | originatorApplicationInfo | See <OriginatorApplicationInfo> type | Information |  |   {  “subscriptionId”: string,  “originatorUuid”: string,  “originatorContactRoleCd”: string,  “contactMechanismTypeCd”: string,  “contactOptinTypeCd”: string,  “billingAccountTypeCd”: string,  “effectiveStartDt”: date,  “effectiveEndDt”: date,  “originatorApplicationInfo”: <OriginatorApplicationInfo>  }  Where OriginatorApplicationInfo is described as  OriginatorApplicationInfo   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Datatype** | **Description** | **Possible/typical values** | | userId | string | User or agent |  | | applicationId | string | Application id |  | | salesRepId | string | Sales rep id |  | | channelOrganizationId | string | Channel org id |  | | outletId | string | Outlet id |  | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessCustomerPreferenceManagementService\_v2\_0**  retrieveSubscriberNotificationAlertPreferenceHistory | 2000 ms average; 5000 ms peak |   Expected average response time: 2300 ms |

## notification-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/notification-history | |
| **Method** | GET |
| **Description** | Retrieve notification history details such as creation time, SMS or email etc. |
| **XSD reference** | Refer to element: getNotificationHistory and getNotificationHistoryResponse |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | sub | “4165551212” | N | Pass in subscriber to refine search | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | EnterpriseCustomerNotificationInquiryService call failed | Underlying error code | Underlying error message | Any other Policy or Service Exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “notificationHistoryResponse”:  {  “notificationHistoryList”: [ < NotificationHistory > ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | notificationHistoryList | Array of < NotificationHistory > | Notification history |  | | status | <status> | status |  |   Where NotificationHistory is described as  NotificationHistory   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionId | string | Subscription id | “203343421” when applicable | | contactEventTypeCd | string | Contact event type  RefPDS: CNTCT\_EVENT\_TYPE | “1” – Usage Threshold alert is only supported | | recipientContactRoleCd | string | Destination recipients role,  Contact role  RefPDS: CONTACT\_ROLE | DVC - device,  ACCTMBR - account member,  ACCTMGR - account manager,  ACCTOWNR – account owner | | contactMechanismTypeCd | string | Contact mechanism type  RefPDS: CNTCT\_MECHANISM\_TYPE | SMS, EMAIL | | createDt | date | Create time | e.g. “2013-06-07T08:50:44” | | destinationAddressTxt | string | Destination address | sms number or email address | | originatingAddressTxt | string | Originating address | originating system or identity of sms or email origination | | deliveryStatusCd | string | Overall contact delivery event status – indicates whether event was successfully initiated  RefPDS: CNTCT\_EVENT\_STAT | COMPLETED ,  FAILED,  INPROGRESS,  SCHEDULED,  RETRY, | | deliverySubStatusCd | string | Actual EMAIL or SMS delivery status.  SMS RefPDS: SMS\_PROCESS\_STAT  EMAIL RefPDS: GCC\_CORRESPONDENCE\_STATUS | SMS values:  UNDELIV,  REJECTD,  EXPIRED,  DELETED,  ALTROUTEDELIVRD,  INBRCVD,  INITIAL ,  SUBMITTING,  ENROUTE,  DELIVRD  EMAIL values:  PENDING,  CANCELLED,  DELIVERED,  IN\_BATCH,  ERROR,  INCOMPLETE,  SENDING | | contentTxt | String | SMS notification content  or  Email Content Id (i.e. no actual email content can be provided at this time) | SMS e.g. “You've used 90% of your included CDN data and have 20MB left until 2014-12-31. Visit telus.com/myaccount  to track usage and get more info. Free TELUS msg”  Email e.g. “11405664” |   {  “subscriptionId”: string,  “contactEventTypeCd”: string,  “recipientContactRoleCd”: string,  “contactMechanismTypeCd”: string,  “createDt”: date,  “destinationAddressTxt”: string,  “originatingAddressTxt”: string,  “deliveryStatusCd”: string,  “deliverySubStatusCd”:string,  “contentTxt”: string  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **EnterpriseCustomerNotificationInquiryService** **\_v1\_1**  retrieveConsolidatedNotificationHistory | 1000 ms average; 3000 ms peak |   Expected average response time: 2000 ms |
| **Example** | Example of response:  {"notificationHistoryResponse": {  "notificationHistoryList": [  {  "subscriptionId": "8373072",  "contactEventTypeCd": "1",  "recipientContactRoleCd": "DVC",  "contactMechanismTypeCd": "SMS",  "createDt": "2014-12-19T08:00:53-05:00",  "destinationAddressTxt": "7781811931",  "originatingAddressTxt": "5338",  "deliveryStatusCd": "COMPLETED",  "deliverySubStatusCd": "UNDELIV",  "contentTxt": "You have spent more than $50 on roaming data this bill cycle ending 2015-01-14. Your current rate is $0.75/MB in USA. Visit telus.com/myaccount to track usage and get more info. Free TELUS msg."  },  {  "subscriptionId": "8373072",  "contactEventTypeCd": "1",  "recipientContactRoleCd": "ACCTOWNR",  "contactMechanismTypeCd": "EMAIL",  "createDt": "2014-12-19T06:55:50-05:00",  "destinationAddressTxt": "notification\_1171@ci-qa.com",  "deliveryStatusCd": "COMPLETED",  "deliverySubStatusCd": "DELIVERED",  "contentTxt": "11407361"  }  ],  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  }} |

## block-consent-history

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/sub/{sub}/block-consent-history | |
| **Method** | GET |
| **Description** | Retrieve block and unblock consent history for a given subscriber |
| **XSD reference** | Refer to element: getUsageBlockConsentHistory and getUsageBlockConsentHistoryResponse |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | | techtype | “LTE”, “PCS” | Y | LTE (and HSPA), PCS (CDMA) | | brand | “TELUS”, “KOODO” | Y |  | | application | Example:  “10” - SSF,  “27” – Smartdesktop,  “9958” – SMS,  “8471”, UNE etc | Y | Application id’s to filter on – consumer can choose to filter on.  Application ID can be determined by SRPDS SERVREQ\_APPLICATION\_ID table | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | ServiceRequestInfoService call failed | Underlying error code | Underlying error message | Any other Policy or Service Exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “usageBlockConsentHistoryResponse”:  {  “usageBlockConsentHistoryList”: [ < UsageBlockHistory> ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageBlockHistoryList | Array of <UsageBlockConsentHistory> | Block unblock consent history |  | | status | <status> | status |  |   Where UsageBlockConsentHistory is described as  UsageBlockConsentHistory   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | eventDt | date | Consent event date | e.g. “2013-06-07T08:50:44” | | networkZoneTypeId | string | Network zone type | “domestic” (1 - SRPDS),  “roaming” (2 – SRPDS) | | eventTypeId | string | Event type | “consent\_event” (23 - SRPDS) | | applicationCd | string | Application involved in consent history | See list of application in request | | actorTypeList | Array of <ActorType> | Actors that were involved in consent history | i.e. dealer code = A010000001  i.e. sales rep code = 0000 |   {  “eventDt”: date,  “networkZoneTypeId”: string,  “eventTypeId”: string,  “applicationCd”: string,  “actorTypeList”: [ <ActorType> ],  }  Where ActorType is described as   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | actorTypeCd | string | Actor type, such as agent, subscriber identifier etc | “8” – subscriber  “19” – agent  “3“ – sales rep  etc.  SRPDS table  SERVREQ\_ACTOR\_TYPE | | actorTxt | string | Actor value | “4165551212”, “t345333”, “A001000001” etc | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **ServiceRequestInfoService\_v5\_0**  findServiceRequests | 3000 ms average; 5000 ms peak |   Expected average response time: 4000 ms |

## block-event-history

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/sub/{sub}/block-event-history | |
| **Method** | GET |
| **Description** | Retrieve block and unblock event history for a given subscriber |
| **XSD reference** | Refer to element: getUsageBlockUnblockEventHistory and getUsageBlockUnblockEventHistoryResponse |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | from | e.g. "2013-06-01" | Y | Start date of range | | to | e.g. "2013-06-30" | Y | End date of range | | techtype | “LTE”, “PCS” | Y | LTE (and HSPA), PCS (CDMA) | | brand | “TELUS”, “KOODO” | Y |  | | application | Example:  “10” - SSF,  “27” – Smartdesktop,  “9958” – SMS,  “8471”, UNE etc | Y | Application id’s to filter on – consumer can choose to filter on.  Application ID can be determined by SRPDS SERVREQ\_APPLICATION\_ID table | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | ServiceRequestInfoService call failed | Underlying error code | Underlying error message | Any other Policy or Service Exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “tns:usageBlockEventHistoryResponse”:  {  “usageBlockEventHistoryList”: [ < UsageBlockEventHistory> ],  “status”: <status>  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageBlockEventHistoryList | Array of <UsageBlockEventHistory> | Block unblock event history |  | | status | <status> | status |  |   Where UsageBlockEventHistory is described as:  UsageBlockEventHistory   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | eventDt | date | Block/unblock event date | e.g. “2013-06-07T08:50:44” | | networkZoneTypeId | string | Network zone type | “domestic” (1 - SRPDS)  “roaming” (2 – SRPDS) | | blockConditionTypeCd | string | Block condition such as permanent, threshold, immediate | “threshold” (1 – SRPDS)  “permanent” (2 – SRPDS)  “immediate” (3 – SRPDS) | | thresholdAmt | double | Threshold amount | Returns amount | | thresholdPercentage | double | Threshold percentage | Returns threshold percentage | | eventTypeId | string | Event type | “block\_event” (24 – SRPDS)  “unblock\_event” (25 – SRPDS) | | applicationCd | string | Application involved in consent history | See list of application in request | | actorTypeList | Array of <ActorType> | Actors that were involved in consent history |  |   {  “eventDt”: date,  “networkZoneTypeId”: string,  “blockConditionTypeCd”: string,  “eventTypeId”: string,  “applicationCd”: string,  “actorTypeList”: [ <ActorType> ],  }  Where ActorType is described as   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | actorTypeCd | string | Actor type, such as agent, subscriber identifier etc | “8” – subscriber  “19” – agent  “3“ – sales rep  etc.  SRPDS table  SERVREQ\_ACTOR\_TYPE | | actorTxt | string | Actor value | “4165551212”, “t345333” |   {  “actorTypeCd”: string,  “actorValueTxt”: string  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **ServiceRequestInfoService\_v5\_0**  findServiceRequests | 3000 ms average; 5000 ms peak |   Expected average response time: 4000 ms |

# PaymentManagement 3.0 REST Service

This service provides operations in support of managing wireless and wireline (FFH) payments – e.g.  setting up or disabling preauthorized payments via credit card or chequing account.

## payment-info

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/payment-info | |
| **Method** | GET |
| **Description** | Account info related to billing – amount due, past due, next bill, current payment method (PAP via credit card or bank account), etc. Typically displayed at the start of a payment flow. Supports wireless and wireline. |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system – default is wireless (initially only KB will be supported) | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN – we'll catch this and call it a 400 | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  <accountInfo>,  "status" : <status>  where **accountInfo** is composed of:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | ban | String | Account number | 78645443 | | billCycleInfo | <cycle info> | Bill cycle info as defined above in UsageManagement – bill-cycle-info | 2013-06-13T00:00:00 | | typeTxt | string | Account type – single character | "I", "B", "C" | | subTypeTxt | string | Account subtype – single character | "R", "Q", "1" | | brandId | int | Brand ID | 1 – Telus; 3 – Koodo | | activationDt | date | Account activation date | 2013-06-14T00:00:00 | | accountStatusCd | string | Status code | "O" (open"), "S" (suspended) | | paymentMethodCd | string | Payment method | "regular", "credit", "cheque" | | cardType | string | Type of Credit Card | "VI", "MC", "AM" \*\*v2.1 (WLN only) | | creditCardLast4Txt | string | Last 4 digits of registered card | "4577" | | creditCardMonthTxt | string | Expiry month – 2 char | "05" | | creditCardYearTxt | string | Expiry year – 2 char | "03" | | bankCodeTxt | string | Bank institution code | "003" | | bankAccountTxt | string | Bank account number | "340-4544" | | bankTransitNumTxt | string | Bank branch (transit) number | "04034" | | dueAmt | double | Amount due | 16.95 | | dueDt | date | Date payment due | 2013-06-14T00:00:00 | | pastDueAmt | double | Amount past due | 37.5 | | lastPaymentAmt | double | Amount of last payment | 23.45 | | lastPaymentDt | date | Date of last payment | 2013-06-14T00:00:00 | | hasSpendingLimitInd | boolean | Is account on spending limit program | Y, N | | spendingLimitAmt | Int | Spending limit cap | 200 | | maxDebitAmt | double | Maximum amount to withdraw – WLN only (v2.1) | 100.00 |   Example:  {  "accountNum": "22478864",  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": null,  "startDateTxt": null,  "daysElapsed": 27,  "daysInCycle": 31,  "unitTxt": null,  "billCycleCd": 22,  "captionTxt": null,  "endsTxt": null  },  "typeTxt": "I",  "subTypeTxt": "R",  "brandId": 1,  "activationDt": "2012-12-15T00:00:00",  "accountStatusCd": "O",  "paymentMethodCd": "credit",  "creditCardLast4Txt": "1755",  "creditCardMonthTxt": "07",  "creditCardYearTxt": "14",  "bankCodeTxt": null,  "bankAccountTxt": null,  "bankTransitNumTxt": null,  "dueAmt": 16.95,  "dueDt": "2013-07-15T00:00:00",  "pastDueAmt": 0.0,  "lastPaymentAmt": 16.95,  "lastPaymentDt": "2013-06-18T00:00:00",  "hasSpendingLimitInd": "N",  "spendingLimitAmt": 0,  "status" :  { "statusCd" : "200",  "statusSubCd" : null,  "statusTxt" : "OK",  "systemErrorTimeStamp": null,  "systemErrorCd" : 0,  "systemErrorTxt" : null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## payment-method

|  |  |
| --- | --- |
| **OPERATION**  account/payment-method | |
| **Method** | PUT |
| **Description** | Set payment method – preauthorized credit card, preauthorized chequing (bank account), or regular (not preauth). Supports wireless and wireline. |
| **Input** | where body of Request is:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Default is wls | | ban | 4367776 | Mandatory | Account number | | paymentMethodCd | "creditcard", "bankaccount", "regular" | Mandatory | "regular" cancels preauthorized cc/bank | | cardType | “VI” | Optional | Credit Card type like "VI", "MC", "AM"  \*\*v2.1 | | cardTokenNum | 342343 | Optional\* | Credit card token from Avalon  \* mandatory if method="creditcard" | | cardHolderTxt | “Fname Lname” | Optional | Name on Credit Card \*\*v2.1 | | cardFirst6Num | 451902 | Mandatory | First 6 digits of CC number | | cardLast4Num | 3455 | Mandatory | Last 4 digits of CC number | | auxiliaryPaymentInfoCd | j6ou21TZmyc | Optional | \*\*v2.1 | | expiryMonthNum | "02", "12" | Optional\* | Expiry month in MM format  \* mandatory if method="creditcard" | | expiryYearNum | "14", "16" | Optional\* | Expiry year in YY format  \* mandatory if method="creditcard" | | bankNum | 003 | Optional\* | Institution number  \* mandatory if method="bankaccount" | | bankTransitNum | 0544 | Optional\* | Transit number  \* mandatory if method="bankaccount" | | bankAccountNum | 7014445 | Optional\* | Bank account number  \* mandatory if method="bankaccount" | | maxDebitAmt | 100.00 | Optional\* | Maximum amount to withdraw – WLN only | | notificationInd | true | Optional | Sends Notification email if true. By default notification will be sent unless specified as false. | | appTrail | List of <app-info> | Optional\* | \* mandatory if method="creditcard" | |  |  |  |  |   where **app-info** is:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | applicationId | 12685 | Mandatory | 12685 is CMDB ID for USS | | userId | myloginid | Mandatory | Login id from app | | ipAddress | 177.204.44.17 | Mandatory | IP address |   The parameters will be passed in the body of the HTTP request within a JSON structure, e.g.   |  |  |  | | --- | --- | --- | | creditcard | bankaccount | regular | | {"system" : "wls",  "ban" : "4367776",  "paymentMethodCd":"creditcard",  "cardTokenNum" : "345AZ45449",  "expiryMonthNum" : "02",  "expiryYearNum" : "14",  “ notificationInd” : true,  "appTrail" : [{  "appId" : "12685",  "userId": "myloginid",  "ipAddress": "274.20.37.1"  }]  } | {"system" : "wls",  "ban" : "4367776",  "paymentMethodCd":"bankaccount",  "bankNum" : "007",  "bankBranchNum" : "0544",  "bankAccountNum " : "343459",  } | {  "system" : "wls",  "ban" : "4367776",  "paymentMethodCd":"regular",  } |   NOTE: The token representing the credit card must be generated by calling the Avalon EncryptionService. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | Authorization-number=xxx, reference-number=yyy | Global-payments-response-code=zzz | global-payments-response-text=aaaaa |  | | 400 | ICCI | Credit card information is required for a credit card payment type |  |  | Missing cc token, expiration date info | | 400 |  | Invalid first 6 digits of credit card number |  |  | First 6 digits parameter not passed or not 6 digits when payment method = "creditcard" | | 400 |  | Invalid last 4 digits of credit card number |  |  | Missing or invalid 4 digits passed. | | 400 |  | Audit trail is required to handle credit card information. |  |  |  | | 400 |  | IP address in audit trail not valid |  |  |  | | 400 |  | appInfo.userId may not be null |  |  |  | | 400 |  | appInfo.applicationId may not be null |  |  |  | | 400 |  | Invalid bank code. In Canada bank codes are 3-digit integers. |  |  |  | | 400 |  | Invalid branch number. In Canada branch numbers are 5-digit integers. |  |  |  | | 400 |  | Invalid account number. Account Number should be 1 or more integers |  |  |  | | 400 | IBAI | Bank account information is required for a preauthorized debit payment type |  |  |  | | 400 | AF | WlsPaymentService returned error | PYMT17001: PAYMENT\_AVS\_ADDRESS\_IS\_MISSING    PYMT17002:  PAYMENT\_AVS\_GP\_ADDRESS\_INVALID  PYMT17082: PAYMENT\_AVS\_TU\_ADDRESS\_INVALID |  | Address Fail 300 | | 400 | ICF | WlsPaymentService returned error | PYMT17070: PAYMENT\_INTERNATIONAL\_CARD |  | International Card  Fail 102 | | 500 |  | Update payment method failed | Global-payments-status-code=zzz | global-payments-status-text=aaaaa | PolicyException error code= 100 | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | transactionId | String (alphanumeric) |  | optional | | status | status | Standard status object |  |   Example:  {  "transactionId" : "123ABC",  "status" :  {  "statusCd" : "200",  "statusTxt" : "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  updatePaymentMethod | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** | n/a |

# PaymentProcessing REST Service

This service provides operations in support of processing payments – e.g. one time bill payment with registered or non-registered credit card.

## payment-with-registered-card

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/payment-with-registered-card | |
| **Method** | POST |
| **Description** | Make a payment against an account balance using registered credit card \*\* NOTE: currently only Wireless supported \*\* |
| **Auth Level** | OWNER-MANAGER |
| **URI** | /cmo/billinginquirymgmt/paymentprocessing/account/{ban}/payment-with-registered-card |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system – default is wireless (currently only wireless supported) | | ban | Account number | Mandatory |  | | customerId | customer ID | Optional\* | Mandatory if system="ffh" | | paymentAmt | Dollar amount in double format | Mandatory | The payment amount | | appTrail | List of <appInfo> | Mandatory |  |   where appInfo is:   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | applicationId | ID of calling app, e.g. “USS” | Mandatory | If possible, use CMDB ID. | | userId |  | Mandatory | some user ID, e.g. login ID | | ipAddress | 177.204.44.17 | Mandatory | IP address of user |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "system": "wls",  "ban": "3420778",  "paymentAmt": 25.50,  "appTrail": [{  "applicationId": "USS",  "userId": "caprilfr@rogers.com",  "ipAddress": "203.45.207.8"  }]  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK – $X.XX payment made – charged to credit card XXXXXXXX1234 |  |  |  | | 400 | IB | Invalid BAN | BAN [90653242] Not Found |  |  | | 400 | IC | Invalid Customer ID |  |  |  | | 400 |  | Audit trail is required to handle credit card information. |  |  | No audit trail provided. | | 400 |  | IP address in audit trail not valid |  |  | IP address invalid or not provided within appInfo | | 400 |  | appInfo.userId may not be null |  |  | UserId not provided within appInfo | | 400 |  | appInfo.applicationId may not be null |  |  | App ID not provided within appInfo | | 400 | IAMT | Invalid amount |  |  | Amount can't be 0 | | 400 |  | Validation error: payment amount may not be negative |  |  | Payment amount must be positive | | 400 | OVERPMT | Amount exceeds 120% of amount due |  |  | Maximum 20% overpayment allowed | | 400 | NOBAL | No balance owing |  |  |  | | 500 | ICCT | Invalid cc token |  |  |  | | 500 | PF | Payment failed | PaymentMgmtSvcException.  getErrorCode | PaymentMgmtSvcException.  getErrorCause (first few lines) |  | | 500 | CNR | Payment method or credit card is missing. |  |  | Account is not set up for PAP – can't use this method for payment. | | 500 |  | Other downstream error |  |  |  | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | accountNum | String | Account number | 78645443 | | paidAmt | Double | Amount paid | 47.82 | | cardLast4Txt | string | Last 4 digits of credit card | 5677 | | cardExpiryMonthTxt | string | Month in MM format | 07 | | cardExpiryYearTxt | string | Year in YY format | 13 | | transactionIdTxt | string | Authorization number returned by API | 342334-4 | | status | Status | Standard status footer |  |   Example:  {  "accountNum" : "22478864",  "paidAmt" : 37.50,  "cardLast4Txt" : "5677",  "cardExpiryMonthTxt" : "07",  "cardExpiryYearTxt" : "13",  "transactionIdTxt" : "342334-4",  "status" :  { "statusCd" : "200",  "statusSubCd" : null,  "statusTxt" : "OK",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd" : 0,  "systemErrorTxt" : null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **Account (Client API)**  payBill |  |   Expected average response time: 6000 ms (allowing for time to connect to Client API) |
| **Sample GUI** |  |

## payment-with-unregistered-card

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/payment-with-unregistered-card | |
| **Method** | POST |
| **Description** | Apply payment to account with provided credit card – \*\* card must have same billing name/address as account \*\* |
| **Auth Level** | OWNER-MANAGER |
| **URI** | /cmo/billinginquirymgmt/paymentprocessing/account/{ban}/payment-with-unregistered-card |
| **Input** | Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory** | | **Comments** | |  |  | **Wls** | **FFH** |  | | system | "wls", "ffh" |  |  | Default is wls | | ban | 4367776 | x | x | Account number | | customerId | customer ID |  | x | Mandatory if system="ffh" | | cardTokenNum | 342343 | x | x | Credit card token from Avalon | | auxiliaryPaymentInfoCd | j6ou21TZmyc |  |  | CVV token from Avalon | | cardType | "VI", "MC", "AM" |  |  | We'll attempt to identify type based on first 4 digits if not provided | | cardFirst6Num | 451902 | x | x | First 6 digits of CC number | | cardLast4Num | 3455 | x | x | Last 4 digits of CC number | | expiryMonthNum | "02", "12" | x | x | Expiry month in MM format | | expiryYearNum | "2014", "2016" | x | x | Expiry year in YYYY format | | cardHolderTxt | “fname lname” |  |  | Name on the card | | ~~holderFirstNameTxt~~ |  |  |  | Deprecated | | ~~holderLastNameTxt~~ |  |  |  | Deprecated | | ~~postalCd~~ |  |  |  | Deprecated | | paymentAmt | 47.50 | x | x | Payment amount | | appTrail | List of <appInfo> | x | x | see previous operation |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "system" : "wls",  "ban" : "4367776",  "cardTokenNum" : "345AZ45449",  "auxiliaryPaymentInfoCd" : "4534662A35454",  "cardType" : "VI",  "cardFirst6Num" : "451902",  "cardLast4Num" : "3346",  "expiryMonthNum" : "02",  "expiryYearNum" : "14",  "cardHolderTxt" : "John Doe",  "paymentAmt" : 25.50,  "appTrail" …  }  NOTE: The token representing the credit card must be generated by calling the Avalon EncryptionService. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK – $X.XX payment made – charged to credit card XXXXXXXX1234 |  |  |  | | 400 |  | Audit trail is required to handle credit card information. |  |  | No audit trail provided. | | 400 |  | IP address in audit trail not valid |  |  | IP address invalid or not provided within appInfo | | 400 |  | appInfo.userId may not be null |  |  | UserId not provided within appInfo | | 400 |  | appInfo.applicationId may not be null |  |  | App ID not provided within appInfo | | 400 | IB | Invalid BAN | BAN [90653242] Not Found |  |  | | 400 | IC | Invalid Customer ID |  |  |  | | 400 | IAMT | Invalid amount |  |  | Amount can't be 0 | | 400 |  | Validation error: payment amount may not be negative |  |  | Payment amount must be positive | | 400 | OVERPMT | Amount exceeds 120% of amount due |  |  | Maximum 20% overpayment allowed; no payment allowed if balance is 0 or credit. | | 400 |  | Invalid CC Type |  |  | If provided, throw error if not one of VI/MC/AM (FFH only) | | 400 | ICCT | Invalid CC Type/unable to determine CC type |  |  | Use algorithm to detect based on 1st 4 digits and throw error if can't be determined. | | 400 |  | Invalid first 6 digits of credit card number |  |  | First 6 digits parameter not passed or not 6 digits when payment method = "creditcard" | | 400 |  | Invalid last 4 digits of credit card number |  |  | Missing or invalid 4 digits passed. | | 400 | AF | WlsPaymentService returned error | PYMT17001: PAYMENT\_AVS\_ADDRESS\_IS\_MISSING    PYMT17002:  PAYMENT\_AVS\_GP\_ADDRESS\_INVALID  PYMT17082: PAYMENT\_AVS\_TU\_ADDRESS\_INVALID |  | Address Fail 300 | | 400 | ICF | WlsPaymentService returned error | PYMT17070: PAYMENT\_INTERNATIONAL\_CARD |  | International Card  Fail 102 | | 500 | PF | Payment failed | FFH: PaymentMgmtSvcException.  getErrorCode  Wls: AIS PolicyException | FFH: PaymentMgmtSvcException.  getErrorCause (first few lines)  Wls: PolicyException |  | | 500 |  | Other downstream error |  |  |  | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | accountNum | String | Account number | 78645443 | | paidAmt | Double | Amount paid | 47.82 | | cardLast4Txt | string | Last 4 digits of credit card | 5677 | | cardExpiryMonthTxt | string | Month in MM format | 07 | | cardExpiryYearTxt | string | Year in YY format | 13 | | transactionIdTxt | string | Authorization number returned by API | 342334-4 | | balanceAmt | Double | If available will be returned for FFH accounts | 100.0 | | status | status | Standard status footer |  |   Example:  {  "accountNum" : "22478864",  "paidAmt" : 37.50,  "cardLast4Txt" : "5677",  "cardExpiryMonthTxt" : "07",  "cardExpiryYearTxt" : "13",  "transactionIdTxt" : "342334-4",  "status" :  { "statusCd" : "200",  "statusTxt" : "OK – payment successful",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **Wireless**  **BillingInquiryService 1.5**  processPayment  **AccountInformationService 3.2** getAccountByAccountNumber  **Wireline**  **ConsumerPaymentManagementSvc 1.0**  makePayment  **ConsumerCustomerManagementSvc 2.3**  getFullCustomerInfo | 1000 ms average; 2000 ms peak |   Expected average response time: 2000 ms |
| **Sample GUI** | n/a |

## card-authorization

|  |  |
| --- | --- |
| **OPERATION**  card-authorization | |
| **Method** | POST  NOTE: while this operation is technically more suited to a GET, POST is used here to prevent the credit card information from being logged as part of the URI. |
| **Description** | perform a credit card authorization check |
| **Auth Level** | CII\_USER |
| **URI** | /cmo/billinginquirymgmt/paymentprocessing/card-authorization |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | cardTokenNum | 342343 | Yes | Credit card token from Avalon | | auxiliaryPaymentInfoCd | j6ou21TZmyc | No | CVV token from Avalon | | cardFirst6Num | 451902 | Yes | First 6 digits of CC number | | cardLast4Num | 3455 | Yes | Last 4 digits of CC number | | expiryMonthNum | 2 digit month "02", "12" | Yes | Expiry month in MM format | | expiryYearNum | 4 digit year e.g. “2018” | Yes | Expiry year in YYYY format | | firstNameTxt |  | No | first name on card | | lastNameTxt |  | Yes | last name on card | | streetNameTxt |  | No |  | | streetNumberTxt |  | No |  | | postalCd | “M5N3L2” | Yes |  | | accountTypeCd | “I” | Yes | Use “I” for prepaid | | accountSubtypeCd | “Q” | Yes | Use “Q” for prepaid | | brandCd | “telus”,”koodo” | Yes |  | | authAmt | 0.01 | No | Default is 0.01 | | appId | Your app ID – e.g. SS\_SVC, TDOPA |  | Drives the set of fraud checks performed. Currently both SS\_SVC and TDOPA are configured for: TransUnion AVS, CVV, International and Negative checks. Also used for audit trail purposes in Avalon. |   Example:  POST …/cmo/billinginquirymgmt/paymentprocessing/card-authorization  {  "cardTokenNum": "345AZ45449",  "auxiliaryPaymentInfoCd": "4534662A35454",  "cardFirst6Num": "451902",  "cardLast4Num": "3346",  "expiryMonthNum": "02",  "expiryYearNum": "2018",  "firstNameTxt": "John",  "lastNameTxt": "Doe",  "streetNumberTxt": "25",  "streetNameTxt": "King",  "postalCd": "M5N3L2",  "accountTypeCd": "I",  "accountSubtypeCd": "Q",  "brandCd": "telus",  "appId": "SS\_SVC"  }  NOTE: The tokens representing the credit card and CVV (“auxiliary payment info”) must be generated by calling ACCC. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid first 6 digits of credit card number |  |  | Missing or invalid 6 digits passed. | | 400 |  | Invalid last 4 digits of credit card number |  |  | Missing or invalid 4 digits passed. | | 400 | AF | CardPaymentService returned error | PYMT17001: PAYMENT\_AVS\_ADDRESS\_IS\_MISSING    PYMT17002:  PAYMENT\_AVS\_GP\_ADDRESS\_INVALID  PYMT17082: PAYMENT\_AVS\_TU\_ADDRESS\_INVALID |  | Address Fail 300 | | 400 | ICF | CardPaymentService returned error | PYMT17070: PAYMENT\_INTERNATIONAL\_CARD |  | International Card  Fail 102 | | 400 | BKC | CardPaymentService returned error | PYMT17003 |  | Credit Card in the black list | | 500 |  | CardPaymentService failed | Underlying error code | Underlying error message | Any Policy or Service Exception | | 500 |  | Other downstream error |  |  |  | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | resultCd | String | result code | “SUCCESS”, “FAILURE” | | approvalCd | string | approval code for success/failure | 47.82 | | approvalTxt | sting | message related to success/failure | “CALL FOR AUTH                C”,  “APPROVED 100640” | | transactionIdTxt | string | Authorization number returned by API | 342334-4 | | status | status | Standard status footer |  |   Example:  {  "resultCd": "SUCCESS",  "approvalCd": "164711",  "approvalTxt": "APPROVED 100640",  "transactionIdTxt": "342334",  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **CardPaymentService**  processTransaction |  |   Expected average response time: |
| **Sample GUI** | n/a |

# PrepaidPaymentManagement REST Service

This service provides operations in support of making prepaid topup and auto topup payments.

## prepaid-info

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/prepaid-info | |
| **Method** | GET |
| **Description** | Get prepaid balance info – balance, autotopup info |
| **Input** | **ban** – account number |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN |  | BAN [2223334] Not Found | AIS fails due to invalid BAN | | 400 | NP | Not prepaid account |  | BAN [70746204] is not Prepaid account | Account type/subtype not equal to I/Q | | 400 | BR | validation errors: … |  |  | Any request validation errors | | 500 |  | AccountInformationService call failed |  |  | Other exceptions from downstream domain service | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | balanceAmt | double | Prepaid balance | 34.4 | | balanceExpiryDate | date |  |  | | minimumBalanceDate | date |  |  | | billingTypeCd | int |  |  | | airtimeRateAmt | double | Current airtime rate |  | | longDistanceRateAmt | double | Current LD rate |  | | outstandingChargeAmt | double |  |  | | maximumBalanceCapAmt | double |  |  | | topupCreditCardLast4 | String |  |  | | topupCreditCardExpiryMonth | String |  |  | | topupCreditCardExpiryYear | String |  |  | | autoTopUpChargeAmt | Int |  |  | | autoTopUpNextChargeDt | date |  |  | | autoTopUpHasThresholdInd | boolean | Indicates whether is on threshold topup |  | | autoTopUpThresholdAmt | int | Balance below which autotopup is triggered (typically $5) | 5 | | status | Status |  |  |   Example:  {  "balance": 38.5,  "balanceExpiryDate": "2013-06-24T13:45:03-4:00",  "minimumBalanceDate": "2013-06-24T13:45:03-4:00",  "billingTypeCd": 1,  "airtimeRateAmt": 0.29,  "longDistanceRateAmt": "0.45",  "outstandingChargeAmt": null,  "maximumBalanceCapAmt": 150.0,  "topupCreditCardLast4": "1644",  "topupCreditCardExpiryMonth": "06",  "topupCreditCardExpiryYear": "14",  "autoTopUpChargeAmt": 25,  "autoTopUpNextChargeDt": "2013-07-24T00:00:00-4:00",  "autoTopUpHasThresholdInd": true,  "autoTopUpThresholdAmt": 5.0,  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK – payment successful",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## topup-settings

|  |  |
| --- | --- |
| **OPERATION**  topup-settings | |
| **Method** | GET |
| **Description** | Reference data useful for supporting prepaid top-up flows – valid top-up amounts, threshold amount |
| **Input** | Parameters  No input parameters – returns reference data  Example:  <domain>/v1/cmo/selfmgmt/prepaidpaymentmanagement/topup-settings |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | SubscriptionService call failed |  |  | Any SS downstream error | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | oneTimeTopUpAmtList | list of double | list of valid amounts for one-time top-ups | 10.0, 25.0, 50.0, 100.0 | | autoTopUpAmtList | list of double | list of valid amounts for auto-topups | 10.0, 25.0, 50.0 | | thresholdAmtList | list of double | list of valid threshold amounts for auto-topups | 5.0 – note – currently 5 is the only valid value but it’s possible in the future other values will be introduced – by default you should take the first one | | status | Status |  |  |   Example:  {  "oneTimeTopUpAmtList": [  10.0, 25.0, 50.0, 100.0  ],  "autoTopUpAmtList": [  10.0, 25.0, 50.0  ],  "thresholdAmtList": 5.0,  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **ProductOfferingService v. 2.2**  getBalanceRechargeDenominationsByType  getBalanceRechargeThresholdDenominations | 500 ms |   Expected average response time: 500 ms |
| **Sample GUI** |  |

## prepaid-balance (POST)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/prepaid-balance | |
| **Method** | POST |
| **Description** | Top-up |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | amount | Dollar amount in int format | \*Optional | The top-up amount – 10, 25, 50, 100 – mandatory if topUpType is “creditcard” | | topUpType | "aircard", "creditcard" | Mandatory |  | | pin | Aircard PIN | \*Optional | Mandatory if topUpType is "aircard" | | appTrail | List of app-info | Mandatory | see payment-with-registered-card for example |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  Aircard top-up:  {  "topUpType" : "aircard",  "pin" : "3448900000988",  "appTrail": [ {  "applicationId": "3423",  "userId": "oz@hotmail.com",  "ipAddress": "203.45.207.8"  }]  }  Credit card top-up:  {  "amount" : 25,  "topUpType" : "creditcard",  "appTrail": [ {  "applicationId": "3423",  "userId": "oz@hotmail.com",  "ipAddress": "203.45.207.8"  }]  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid Subscriber Phone Number | PPS-PE-PPS100 | Subscriber Not Found | SIS fails due to invalid sub | | 400 | IP | Invalid PIN | Eg.   * PPS-PE-V101 * PPS-PE-V105 * PPS-PE-V199 | Eg.   * Voucher Locked * Voucher Stolen * Unknown Voucher Status | Generic Invalid PIN error | | 400 | EP | PIN is expired | PPS-PE-V106 | Voucher Expired | Expired PIN | | 400 | UP | PIN already in use | PPS-PE-V103 | Voucher in Use | PIN in use | | 400 | IC | Invalid credit card | Eg.   * PPS-PE-C101 * PPS-PE-C102 * PPS-PE-C103 | Eg.   * Invalid Credit Card Number * Credit Card Refused by Financial Institution * Maximum Credit Exceeded on Credit Card | Invalid CC registrered on the account | | 400 | CNR | Credit card not registered on the account | PPS-PE-C107 | Subscriber Does Not Have a Registered Credit Card |  | | 400 | ECCTL | Exceeded daily credit card top-up limit | PPS-PE-C104 | Exceeded Daily Cumulative Top-up Limit |  | | 400 | BR | validation errors: |  |  | Any request validation errors | | 500 |  | SubscriptionManagementService call failed |  |  | Any SIS downstream error | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | balanceAmt | Double | Updated prepaid balance | 34.4 | | aircardAmt | Double | Value of airtime card | 25.0, 50.0, null (for cc) | | referenceNum | String | Number returned from successful cc transaction |  | | status | Status |  |  |   Example:  {  "balanceAmt ": 38.5,  "airCardAmt": 25.0,  "referenceNum" : 123425,  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **SubscriptionManagementService v 3.0**  rechargeBalanceByVoucher  rechargeBalanceByRegisteredPaymentInstrument | 2000 ms average; 5000 ms peak |   Expected average response time: 2200 ms |
| **Sample GUI** |  |

## prepaid-auto-topup (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/prepaid-auto-topup | |
| **Method** | PUT |
| **Description** | Auto Top-up |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | rechargeType | "interval", "threshold" | Mandatory | “**interval**” indicates auto top-up every 30 days; “**threshold**” also adds a top-up whenever balance drops below the threshold amount, which defaults to $5 if not specified below. | | rechargeAmt | Dollar amount in int format  Valid values are 10, 25, 50 | Mandatory |  | | thresholdAmt | Dollar amount in int format, typically 5 | Optional | When the client’s balance drops below this amount an auto top-up will be triggered. If not supplied, defaults to $5. | | isImmediateInd |  | Optional | Indicates whether the first amount is charged immediately. Defaults to false if not supplied. Don’t set it for auto top-up change. | | appTrail | List of app-info | Mandatory | see payment-with-registered-card for example |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "rechargeType" : “threshold”,  "rechargeAmt" : 25,  "thresholdAmt" : 5,  "isImmediateInd" : true,  "appTrail": [ {  "applicationId": "3423",  "userId": "oz@hotmail.com",  "ipAddress": "203.45.207.8"  }]  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid Subscriber Phone Number | PPS-PE-PPS100 | Subscriber Not Found | SIS fails due to invalid sub | | 400 | IC | Invalid credit card | Eg.   * PPS-PE-C101 * PPS-PE-C102 * PPS-PE-C103 | Eg.   * Invalid Credit Card Number * Credit Card Refused by Financial Institution * Maximum Credit Exceeded on Credit Card | Invalid CC registrered on the account | | 400 | CNR | Credit card not registered on the account | PPS-PE-C107 | Subscriber Does Not Have a Registered Credit Card |  | | 400 | ECCTL | Exceeded daily credit card top-up limit | PPS-PE-C104 | Exceeded Daily Cumulative Top-up Limit |  | | 400 | BR | validation errors: |  |  | Any request validation errors | | 500 |  | SubscriptionManagementService call failed |  |  | Any SIS downstream error | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | balanceAmt | double | Updated prepaid balance | 34.4 | | status | Status |  |  |   Example:  {  "balanceAmt ": 38.5,  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **SubscriptionManagementService v 3.0**  registerAutomaticRecharge  changeAutomaticRecharge | 2000 ms average; 5000 ms peak |   Expected average response time: 2200 ms |
| **Sample GUI** |  |

## payment-card (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/payment-card | |
| **Method** | PUT |
| **Description** | Registers card or updates existing registered credit card |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | cardTokenNum | 100000000000004025888 | Mandatory | Credit card token from Avalon | | cardTypeCd | "VI", "MC", "AM" | Mandatory | We'll attempt to identify type based on first 4 digits if not provided | | cardFirst6Num | 451902 | Mandatory | First 6 digits of CC number | | cardLast4Num | 3455 | Mandatory | Last 4 digits of CC number | | expiryMonthNum | "02", "12" | Mandatory | Expiry month in MM format | | expiryYearNum | "14", "16" | Mandatory | Expiry year in YY format | | appTrail | List of app-info | Mandatory | see payment-with-registered-card for example |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.,  {  "cardTypeCd": "VI",  "cardTokenNum": "100000000000004025888",  "cardFirst6Num": "451902",  "cardLast4Num": "3455",  "expiryMonthNum": "02",  "expiryYearNum": "16"  "appTrail": [ {  "applicationId": "3423",  "userId": "oz@hotmail.com",  "ipAddress": "203.45.207.8"  }]} |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid Subscriber Phone Number | PPS-PE-PPS100 | Subscriber Not Found | SIS fails due to invalid sub | | 400 | IC | Invalid credit card | Eg.   * PPS-PE-C101 * PPS-PE-C102 * PPS-PE-C103 | Eg.   * Invalid Credit Card Number * Credit Card Refused by Financial Institution * Maximum Credit Exceeded on Credit Card | Invalid CC | | 400 | BR | validation errors: |  |  | Any request validation errors (valid cardtype, all digits, max length, valid expiry month/year format etc.) | | 500 |  | SubscriptionManagementService call failed |  |  | Any SIS downstream error | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | sub | String |  |  | | cardType | String |  |  | | cardFirst6Num | String |  |  | | cardLast4Num | String |  |  | | expiryMonthNum | String |  |  | | expiryYearNum | String |  |  | | status | Status |  |  |   Example:  {  "sub": "4165468888",  "cardTypeCd": "VI",  "cardFirst6Num": "451902",  "cardLast4Num": "3455",  "expiryMonthNum": "02",  "expiryYearNum": "16",  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **SubscriptionManagementService v 3.1**  registerAutomaticRecharge  changeAutomaticRecharge | 1000 ms average; 2000 ms peak |   Expected average response time: 2200 ms |
| **Sample GUI** |  |

# AccountMgmt REST Service

This service provides operations in support of managing the account. Initially this will support wireless only (including those accounts consolidated with FFH accounts).

## account-pin

|  |  |
| --- | --- |
| **OPERATION**  account/account-pin | |
| **Method** | PUT |
| **Description** | Update the account pin number |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | customerID | Used by FFH | \*Optional | \*Mandatory for FFH | | account | Account number | \*Optional | \*Mandatory for Wireless | | accountTypeCode | Account type code | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | accountSubTypeCode | Account subtype code | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | sourceSystemId | Billing system source id | \*Optional | Identifies wireless/wireline  \*Mandatory if using account | | brandId | Brand id 1 for telus 3 for Koodo | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | userId | User id of user logging in | Mandatory | This is used for audit purposes. Use sourceSystemId when user id is unknown. | | originalPIN | Original PIN code | Mandatory | Original PIN code is used for verification purposes | | newPIN | New PIN code to be saved | Mandatory | New PIN to replace original PIN |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "account" : "3420778",  "accountTypeCode" : "I",  "accountSubTypeCode" : "R",  "sourceSystemId" : "130",  "brandId" : 1,  "userID" : "130",  "originalPIN" : "1234",  "newPIN" : "4321"  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN – we'll catch this and call it a 404 | | 400 |  | Invalid Input parameter | ConsumerIdentityProfileMgmtSvc.  validation.error |  | Returned when input parameters fail validation. | | 400 |  | Invalid account type (not prepaid) |  |  | prepaidAccountInfo is null; also account type/subtype not equal to I/Q | | 500 |  | Failed to update Owner Credentials | ConsumerIdentityProfileMgmtSvc.  UpdateOwnerIdentityCredential.failed |  | Returned when there is a failure while trying to update the identity credential. | | 500 |  | Failed to get Owner Credentials | ConsumerIdentityProfileMgmtSvc.  GetOwnerIdentityCredentialByType.failed |  | Returned when there is a failure while trying to retrieve the identity credential. | | 500 |  | Failed to update PIN | WirelessAccountMgmtService.  updateAccountPassword.failed |  | Returned when there is a failure while trying to update the wireless account PIN in KB. | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example:  {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK – update pin successful",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **ConsumerIdentityProfileManagementService v 1.0**  updateOwnerIdentityCredential | 500 ms average; 1500 ms peak |   Expected average response time: 800 ms |
| **Sample GUI** |  |

## billing-address (GET)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/billing-address | |
| **Method** | GET |
| **Description** | Get the account billing address |
| **Input** | **ban** – account number (currently only wireless supported – including those consolidated with an FFH account)  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system – default is wireless (initially only KB will be supported – including KB BANs consolidated with an FFH BAN) | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | AM-B | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN – we'll catch this and call it a 404 | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | status | Status |  |  |   Example:  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "poBox" : null,  "rrNum" : null,  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## valid-address

|  |  |
| --- | --- |
| **OPERATION**  valid-address | |
| **Method** | GET |
| **Description** | Validate an address |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | streetNum |  | Optional | Map to civicNumber | | streetName |  | Optional | Map to streetName | | streetType |  | Optional | Map to streetTypeCode | | streetDirection |  | Optional | Map to streetDirectionCode | | unitTypeCd |  | Optional | Map to unitTypeCode | | unitNum |  | Optional | Map to unitNumber | | poBox |  | Optional | Map to postOfficeBoxNumber | | rrNum |  | Optional | Map to ruralRouteNumber | | cityTxt |  | Mandatory | Map to municipalityName | | provinceCd |  | Mandatory | Map to provinceStateCode | | postalCodeTxt |  | Mandatory | Map to postalZipCode | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  |  |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | validAddressInd | boolean | Indicates whether the address is a valid address or not. | true, false | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | streetType | String | Street type | "ST", "AVE" | | streetDirection | String | Street direction | "N", "E" | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | String | Postal Code | "M5N 2L3" | | addressTypeCd | String |  | "C" - City   "R" - Rural  "P" - P.O. Box  "F" - Foreign | | stationQualifier | String | Used for PO Box | "MAIN" | | stationTypeCd | String | Used for PO Box | "STN" | | countryCd | String | Country | CAN | | status | Status |  |  |   Example:  {  "validAddressInd": "true",  "matchingAddressList": [  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  }  ]  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Services/operation** | **SLA** | | **EnterpriseAddressValidationService v 1.0**  verifyCanadianPostalAddress | 5000 ms average |   Expected average response time: 5500 ms |
| **Sample GUI** |  |

## billing-address (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/billing-address | |
| **Method** | PUT |
| **Description** | Update the billing address on the account |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Mandatory |  | | account |  | Mandatory | If system = wls  Map to knowbilityBillingAccountNumber  If system = ffh  Map to enablerBillingAccountNumber | | validateAddressInd |  | Optional | true/false. Since call is asynchronous the system would not immediately know if the address is in the correct format. If not provided, it will default to N and the address will not be validated. | | streetNum |  | Optional\* | Map to civicNumber, mandatory if address is City | | streetName |  | Optional\* | Map to streetName, mandatory if address is City | | streetType |  | Optional | Map to streetTypeCode | | streetDirection |  | Optional | Map to streetDirectionCode | | unitTypeCd |  | Optional | Map to unitTypeCode | | unitNum |  | Optional | Map to unitNumber | | poBox |  | Optional | Map to postOfficeBoxNumber | | rrNum |  | Optional\* | Map to ruralRouteNumber, mandatory if address is Rural | | cityTxt |  | Mandatory | Map to municipalityName | | provinceCd |  | Mandatory | Map to provinceStateCode | | postalCodeTxt |  | Mandatory | Map to postalZipCode | | countryCd |  | Mandatory | Map to countryCode | | addressTypeCd | "C" - City  "R" - Rural "P" - P.O. Box  "F" - Foreign | Mandatory | Map to addressTypeCode | | stationQualifier | "MAIN" | Optional\* | Mandatory if address is PO Box | | stationTypeCd | "STN" | Optional\* | Map to stationTypeCode, mandatory if address is PO Box |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "system" : "wls",  "account" : "3420778",  "streetNum" : "200",  "streetName" : "Yonge",  "streetType" : "St",  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M1M2M2",  "countryCd" : "CAN",  "addressTypeCd" : "C"  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 200 |  | Invalid address |  |  |  | | 500 |  |  |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example:  {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **EnterpriseBillingAccountManagementService v 1.1**  changeBillingAddress | 4000 ms average; 7000 ms peak |   Expected average response time: 4500 ms |
| **Sample GUI** |  |

# AccountMgmt 2.0 REST Service

This service provides operations in support of managing the account. It is native REST.

## account-pin

|  |  |
| --- | --- |
| **OPERATION**  account/account-pin | |
| **Method** | PUT |
| **Description** | Update the account pin number |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | customerID | Used by FFH | \*Optional | \*Mandatory for FFH | | account | Account number | \*Optional | \*Mandatory for Wireless | | accountTypeCode | Account type code | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | accountSubTypeCode | Account subtype code | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | sourceSystemId | Billing system source id | \*Optional | Identifies wireless/wireline  \*Mandatory if using account | | brandId | Brand id 1 for telus 3 for Koodo | \*Optional | If provided, will bypass call to Account Info service.  \*Mandatory if using account | | userId | User id of user logging in | Mandatory | This is used for audit purposes. Use sourceSystemId when user id is unknown. | | originalPIN | Original PIN code | Mandatory | Original PIN code is used for verification purposes | | newPIN | New PIN code to be saved | Mandatory | New PIN to replace original PIN |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "account" : "3420778",  "accountTypeCode" : "I",  "accountSubTypeCode" : "R",  "sourceSystemId" : "130",  "brandId" : 1,  "userID" : "130",  "originalPIN" : "1234",  "newPIN" : "4321"  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN – we'll catch this and call it a 404 | | 400 |  | Invalid Input parameter | ConsumerIdentityProfileMgmtSvc.  validation.error |  | Returned when input parameters fail validation. | | 400 |  | Invalid account type (not prepaid) |  |  | prepaidAccountInfo is null; also account type/subtype not equal to I/Q | | 500 |  | Failed to update Owner Credentials | ConsumerIdentityProfileMgmtSvc.  UpdateOwnerIdentityCredential.failed |  | Returned when there is a failure while trying to update the identity credential. | | 500 |  | Failed to get Owner Credentials | ConsumerIdentityProfileMgmtSvc.  GetOwnerIdentityCredentialByType.failed |  | Returned when there is a failure while trying to retrieve the identity credential. | | 500 |  | Failed to update PIN | WirelessAccountMgmtService.  updateAccountPassword.failed |  | Returned when there is a failure while trying to update the wireless account PIN in KB. | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example:  {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK – update pin successful",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **ConsumerIdentityProfileManagementService v 1.0**  updateOwnerIdentityCredential | 500 ms average; 1500 ms peak |   Expected average response time: 800 ms |
| **Sample GUI** |  |

## contact-info (GET)

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/contact-info | |
| **Method** | GET |
| **Description** | Get the account billing address and contact phone number(s). **NOTE: this replaces the billing-address GET operation in version 1.** |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system – default is wireless (initially only KB will be supported – including KB BANs consolidated with an FFH BAN) | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | AM-B | Invalid BAN (AIS call failed) | BAN [90653242] Not Found | messageId=APP10004, errorCode=CMB\_AIS\_0001 | AIS fails due to invalid BAN – we'll catch this and call it a 404 | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | homePhoneNum | string | Billing account contact phone | \* Currently wls only | | businessPhoneNum | string | Billing account bus. phone | \* Currently wls only | | businessPhoneExtNum | string | Billing account bus. extension | \* Currently wls only | | status | Status |  |  |   Example:  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "poBox" : null,  "rrNum" : null,  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  "homePhoneNum" : "4168953240"  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.5**  getAccountByAccountNumber  updateContactInformation | 500 ms average; 1000 ms peak | | **ConsumerCustomerMgmtDataGridService 1.3** getCustomerInfoByCustomerIdList |  |   Expected average response time: 1000 ms |
| **Sample GUI** |  |

## valid-address

|  |  |
| --- | --- |
| **OPERATION**  valid-address | |
| **Method** | GET |
| **Description** | Validate an address |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | streetNum |  | Optional | Map to civicNumber | | streetName |  | Optional | Map to streetName | | streetType |  | Optional | Map to streetTypeCode | | streetDirection |  | Optional | Map to streetDirectionCode | | unitTypeCd |  | Optional | Map to unitTypeCode | | unitNum |  | Optional | Map to unitNumber | | poBox |  | Optional | Map to postOfficeBoxNumber | | rrNum |  | Optional | Map to ruralRouteNumber | | cityTxt |  | Mandatory | Map to municipalityName | | provinceCd |  | Mandatory | Map to provinceStateCode | | postalCodeTxt |  | Mandatory | Map to postalZipCode | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  |  |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | validAddressInd | boolean | Indicates whether the address is a valid address or not. | true, false | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | streetType | String | Street type | "ST", "AVE" | | streetDirection | String | Street direction | "N", "E" | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | String | Postal Code | "M5N 2L3" | | addressTypeCd | String |  | "C" - City   "R" - Rural  "P" - P.O. Box  "F" - Foreign | | stationQualifier | String | Used for PO Box | "MAIN" | | stationTypeCd | String | Used for PO Box | "STN" | | countryCd | String | Country | CAN | | status | Status |  |  |   Example:  {  "validAddressInd": "true",  "matchingAddressList": [  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  }  ]  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Services/operation** | **SLA** | | **EnterpriseAddressValidationService v 1.0**  verifyCanadianPostalAddress | 5000 ms average |   Expected average response time: 5500 ms |
| **Sample GUI** |  |

## contact-info (PUT)

|  |  |
| --- | --- |
| **OPERATION**  account/contact-info | |
| **Method** | PUT |
| **Description** | Update the billing address and contact phone number(s) on the account. **NOTE: this replaces the billing-address PUT operation in version 1.** |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Mandatory |  | | account |  | Mandatory | If system = wls  Map to knowbilityBillingAccountNumber  If system = ffh  Map to enablerBillingAccountNumber | | validateAddressInd |  | Optional | true/false. Since call is asynchronous the system would not immediately know if the address is in the correct format. If not provided, it will default to N and the address will not be validated. | | streetNum |  | Optional\* | Map to civicNumber, mandatory if address is City | | streetName |  | Optional\* | Map to streetName, mandatory if address is City | | streetType |  | Optional | Map to streetTypeCode | | streetDirection |  | Optional | Map to streetDirectionCode | | unitTypeCd |  | Optional | Map to unitTypeCode | | unitNum |  | Optional | Map to unitNumber | | poBox |  | Optional | Map to postOfficeBoxNumber | | rrNum |  | Optional\* | Map to ruralRouteNumber, mandatory if address is Rural | | cityTxt |  | Mandatory | Map to municipalityName | | provinceCd |  | Mandatory | Map to provinceStateCode | | postalCodeTxt |  | Mandatory | Map to postalZipCode | | countryCd |  | Mandatory | Map to countryCode | | addressTypeCd | "C" - City  "R" - Rural "P" - P.O. Box  "F" - Foreign | Mandatory | Map to addressTypeCode | | stationQualifier | "MAIN" | Optional\* | Mandatory if address is PO Box | | stationTypeCd | "STN" | Optional\* | Map to stationTypeCode, mandatory if address is PO Box | | homePhoneNum | \*currently wls only | Optional | Only provide if updating (incl. remove) | | businessPhoneNum | \*currently wls only | Optional | Only provide if updating | | businessPhoneExtNum | \*currently wls only | Optional | Only provide if updating |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "system" : "wls",  "account" : "3420778",  "streetNum" : "200",  "streetName" : "Yonge",  "streetType" : "St",  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M1M2M2",  "countryCd" : "CAN",  "addressTypeCd" : "C",  "homePhoneNum" : "4168967450",  "businessPhoneNum" : "6478379852",  "businessPhoneExtNum" : "227"  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 200 |  | Invalid address |  |  |  | | 500 |  | AF |  |  | Address update failure – this is unlikely, as the downstream call is asynchronous | | 500 |  | CF |  |  | Contact info update failure | | 500 |  |  |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | status | Status |  |  |   Example:  {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **EnterpriseBillingAccountManagementService v 1.1**  changeBillingAddress | 4000 ms average; 7000 ms peak | | **AccountInformationService 3.5** updateContactInformation | 500 ms |   Expected average response time: 4500 ms |
| **Sample GUI** |  |

# AccountActivity REST Service

This service provides operations in support of account (and subscriber) activity.

## account/transaction-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/transaction-history | |
| **Method** | GET |
| **Description** | Account-level transaction history – status changes, invoices, payments, changes in payment method, address changes, adjustments (credits) |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | months | e.g. 3 | Optional\* | max months = 6; mandatory if **from** date not passed | | from | 2013-08-07 | Optional\* | Start date for history – ignored if months is supplied; mandatory if **months** not passed | | to | 2013-08-14 | Optional | End date for history – ignored if months is passed; set to current date if not supplied and from is supplied | | transactionType | "status", "invoice", "payment", "payment-method", "address", "adjustment" | Optional | By default, all types of transactions are returned. By passing this in, only the specified type will be returned. | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Invalid date range |  | .getReasonText | HistorySearchException | | 400 |  | Missing or bad parameters - must provide either months or from/to date; months must be <= 6 |  |  |  | | 400 |  | Invalid transaction type |  |  | Must be one of "status", "invoice", "payment", "payment-method", "address", "adjustment" | | 500 |  | XXX History call failed. |  | .getReasonText | TelusAPIException. Indicate which of the several calls failed e.g. getAddressChangeHistory | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  “historyList”: historyItem [],  "status" : <status>  },  where **historyItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | eventDt | Date | Time of event | 2013-06-13T00:00:00 | | eventDateTxt | string | Localized date | June 13 | | transactionTypeCd | string | type | "status", "invoice", "payment", "payment-method", "address", "adjustment" | | transactionTypeTxt | string | Localized transaction desc | "status change", "change of address" | | detailList | detailItem | Array of localized label/desc pairs |  | | Status | Status |  |  |   and **detailItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | labelTxt | string | Localized label | "method of payment", "invoice amount" | | valueTxt | string | Localized value | "Visa Credit Card Payment" |   Example:  {  "historyList": [  {  "eventDt": "2013-06-13T00:00:00",  "eventDateTxt": "June 13",  "transactionTypeCd": "payment",  "transactionTypeTxt": "payment received",  "detailList": [  {  "labelTxt": "method of payment",  "valueTxt": "VISA Credit Card Payment"  }, {  "labelTxt": null,  "valueTxt": "PRE-AUTH CREDIT CARD"  }, {  "labelTxt": "original amount",  "valueTxt": "$21.95"  }, {  "labelTxt": "amount due",  "valueTxt": "$21.95"  }  ]  }, {  "eventDt": "2013-06-10T00:00:00",  "eventDateTxt": "June 10",  "transactionTypeCd": "invoice",  "transactionTypeTxt": "invoice",  "detailList": [  {  "labelTxt": "mailed",  "valueTxt": "yes"  }, {  "labelTxt": "due date",  "valueTxt": "Aug 6"  }, {  "labelTxt": "previous balance",  "valueTxt": "$16.95"  }, {  "labelTxt": "invoice amount",  "valueTxt": "$21.95"  }, {  "labelTxt": "amount due",  "valueTxt": "$16.95"  }  ]  }  ],  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **Client API**  Account.getStatusChangeHistory  Account.getInvoiceHistory  Account.getPaymentHistory  Account.getPaymentMethodChangeHistory  Account.getAddressChangeHistory  Account.getCredits |  |   Expected average response time: 5000 ms |
| **Sample GUI** |  |

## account/payments

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/payments | |
| **Method** | GET |
| **Description** | payments made against an account for a given date range; currently wireless only |
| **Auth Level** | OWNER/MANAGER |
| **Input** | **ban** – account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | system | "wls", "ffh" | Optional | Billing system | | from | e.g. "2013-06-01" | Mandatory | Start date of range | | to | e.g. "2013-06-30" | Optional | End date of range; if not provided default to today | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 500 |  | Payment History call failed. | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  “paymentsResponse”: {  paymentList : [<paymentTransaction>],  "status" : <status>  }  },  where **paymentTransaction** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | paymentDt | Date | Payment date | 2013-06-13 | | paymentAmt | double | Payment amount | 12.45 |   Example:  {  "paymentsResponse" : {  "paymentList" : [ {  "paymentDt" : "2014-02-04",  "paymentAmt" : 15.95  },  {  "paymentDt" : "2014-01-04",  "paymentAmt" : 15.95  }],  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **BillingInquiryService v1.3** (wireless) getPaymentHistory |  | | **PaymentInquiryService (wln API)**  (wireline) |  |   Expected average response time: 1000 ms |
| **Sample GUI** |  |

## sub/transaction-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/transaction-history | |
| **Method** | GET |
| **Description** | subscriber-level transaction history – status changes, invoices, payments, changes in payment method, address changes, adjustments (credits) |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | months | e.g. 3 | Optional | max months = 6; mandatory if **from** date not passed | | from | 2013-08-07 | Optional | Start date for history – ignored if months is supplied; mandatory if **months** not passed | | to | 2013-08-14 | Optional | End date for history – ignored if months is supplied; set to current date if not supplied and from is supplied | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 |  | Invalid date range |  | .getReasonText | HistorySearchException | | 400 |  | Missing parameters - must provide either months or from/to date |  |  |  | | 500 |  | XXX History call failed. |  | .getReasonText | TelusAPIException. Indicate which of the several calls failed e.g. getServiceChangeHistory | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  “historyList”: historyItem [],  "status" : <status>  },  where **historyItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | eventDt | Date | Time of event | 2013-06-13T00:00:00 | | eventDateTxt | string | Localized date | June 13 | | transactionTypeCd | string | type | "priceplan", "service", "contract", "handset", "history", "feature-parameter" | | transactionTypeTxt | string | Localized transaction desc | "status change", "change of address" | | detailList | detailItem | Array of localized label/desc pairs |  | | Status | Status |  |  |   and **detailItem** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | labelTxt | string | Localized label | "method of payment", "invoice amount" | | valueTxt | string | Localized value | "Visa Credit Card Payment" |   Example:  {  "historyList": {  "eventDt": "2013-06-13T00:00:00",  "eventDateTxt": "June 13",  "transactionTypeCd": "feature",  "transactionTypeTxt": "feature add/remove",  "detailList": [  {  "labelTxt": "feature",  "valueTxt": "Canadian LD $0.45"  }, {  "labelTxt": "effective",  "valueTxt": "July 28, 2013"  }  ]  },  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK – update pin successful",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **Client API**  Subscriber.getPricePlanChangeHistory  Subscriber.getServiceChangeHistory  Subscriber.getContractChangeHistory  Subscriber.getHandsetChangeHistory  Subscriber.getHistory  Subscriber.getFeatureParameterHistory |  |   Expected average response time: 5000 ms |
| **Sample GUI** |  |

# PrepaidActivity REST Service

IN PROGRESS

This service provides operations related to prepaid usage.

NOTE: This service is pure REST. See notes under CustomerInfo for how this impacts consumers.

## prepaid-call-and-event-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/prepaid-call-and-event-history | |
| **Method** | GET |
| **Description** | transactional listing of calls and events which can impact prepaid balance |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | months | e.g. 3 | Optional | max months = 6; mandatory if **from** date not passed | | from | 2013-08-07 | Optional | Start date for history – ignored if months is supplied; mandatory if **months** not passed | | to | 2013-08-14 | Optional | End date for history – ignored if months is supplied; set to current date if not supplied and from is supplied | | decorate | y, n | Optional | if y, will provide default descriptions for events | | language | "en", "fr" | Optional | Language of unit descriptions; default is "en" | | cachedetail | y, n | Optional | indicates whether to cache results | | cacherows | 10, 20 | Optional | if cachedetail = "y", populate result with only first x rows from result set; if cacherows not provided, full results are returned along with cache token. If cacherows=0, only the cacheid is returned. | | categoryfilter | “topup”, “adjustment”, “balance”, “feature”, “call”, “in”, “out”, “chargeable”, “free” | Optional | used to filter the list and return a matching subset. In any case, the entire result set will be cached for subsequent calls if cachedetail=y. | | sort | “a”, “d” | Optional | sort order: “a” = by date ascending, “d” = by date descending (default) |   Examples:  **<domain>/prepaidactivity/account/2530433/sub/9058395555/prepaid-call-and-event-history/months=2** – returns all history for last two months without caching the results  **<domain>/prepaidactivity /account/2530433/sub/9058395555/prepaid-call-and-event-history?months=4&cachedetail=y** – returns all history for last 4 months and caches results  **<domain>/prepaidactivity /account/2530433/sub/9058395555/prepaid-call-and-event-history?months=3&cachedetail=y&filter=topup&start=11&rows=20** – caches all history for last 3 months and returns rows 11 thru 30 of a list of all topup-related transactions |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB-AUTH | Account number passed is invalid. Request is invalid. |  |  | BAN not found in datagrid. | | 400 | IS-AUTH | Phone number passed does not correspond to BAN passed. Request is invalid. |  |  | BAN looked up in datagrid, sub doesn't belong to this BAN | | 400 |  | Invalid date range |  |  |  | | 400 |  | Missing parameters - must provide either months or from/to date |  |  |  | | 500 |  | SubscriptionEventHistoryService failed | underlying error code | underlying error message | Any Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  "cacheId" : <cacheId>  "transactionList": <Transaction> [],  "callTransactionList" : <CallTransaction>[],  "eventTransactionList" : <EventTransaction>[],  "status" : <status>  },  where **cacheId** is an alphanumeric string to be used for retrieving a subset of cached data using GET cached-prepaid-call-and-event-history;  where **Transaction** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | transactionDt | Date | Date/Time of event | 2013-06-13T00:00:00Z | | typeCd | String | call or event | “call”, “event” | | categoryCd | String | call/event category – can be used to filter the list | “topup”, “adjustment”, “balance”, “feature”, “call” | | subCategoryCd | string | call subcategory (only for typeCd “call”) | “in”, “out”, “chargeable”, “free” | | beforeBalanceAmt | double | balance prior to transaction |  | | afterBalanceAmt | double | balance upon transaction completion |  | | detailId | integer | pointer to array element of either callTransactionList or eventTransactionList | 1, 2 – e.g. if the itemType = “call”, 3 here means “row 3 (zero-based) in the CallTransactionList” |   and **CallTransaction** is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | callTypeCd | string | type of call | local, long distance, roaming, national, LD US, LD Caribbean, LD International |  | | directionCd | string | inbound/outbound | “in”, “out” |  | | callingPhoneNum | string | number called from |  |  | | calledPhoneNum | string | number called to |  |  | | voiceMailInd | boolean | call to voice mail | true, false |  | | durationNum | int | number of minutes | 1, 10 |  | | localChargeAmt | double | local charges |  |  | | longDistanceChargeAmt | double | LD charges |  |  | | totalChargeAmt | double | total charges |  |  | | rateBreakdownList | <RateBreakdown>[] | breakdown of charges – e.g. for a long distance call, this would have two entries – the regular per minute rate, plus the long distance rate | see below |  | | cityTxt | string | city of call |  |  | | stateTxt | string | state/prov of call |  |  |   and **RateBreakdown** is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | rateCd | string | local, long distance etc. | -7, -6, -5, -4, -3, -2, -1 |  | | rateTxt | string | localized description | “local”, “long distance”, “roaming”, “international”, “US long distance”, “Caribbean long distance”, “national long distance” | x | | rateAmt | double | rate in dollars/cents | 0.5 |  | | rateCountryCd | string | country | “CANADA” |  |   and **EventTransaction** is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | eventTypeCd | int | back end code | -17, -23 |  | | eventTypeTxt | string | short description of code | AC\_TOPUP, AUTORENEW\_CC |  | | eventTypeDescTxt | string | localized description | “credit card topup”, “add feature”, “Account Adjustment”, “disable auto-topup” | x | | confiscatedBalanceAmt | double | can be associated with topup event – the amount originally confiscated due to balance expiry is added back when a top-up occurs within the 7-day grace period. | 35.70 |  | | airtimeCardId | string | airtime card serial number (topup event) | xxxxxxxxxxxx4567 |  | | featureCd | string | soc code associated with feature activation, deactivation, renewal, autorenew, enable/disable autorenew events |  |  | | featureTxt | string | feature description associated with featureCd |  | x | | autoTopupAmt | int | when autotopup is set up or changed, the amount | 10, 25, 50 |  | | adjustmentReasonCd | string | reason for adjustment (for adjustment event) |  |  | | adjustmentReasonTxt | string | localized description of adjustment |  | x | | timeZoneOffsetCd | string | timezone offset of transaction | “ET”, “MT” | x |   Examples  The simplest response would result if cacheresults=y and cacherows=0:  { "cacheId": "100005434534455",  "status":  { "statusCd": 200,  "statusTxt": "OK"  }  }  In order to save a second call (to cached-prepaid-call-and-event-history) when presenting the initial results, one would normally provide the cacherows parameter to bring back the first page of results. Here would be a more typical response, where cacherows=4:  { "cacheId": "100005434534455",  "transactionList": [  {  "transactionDt": "2015-08-20T12:34:00Z",  "itemTypeCd": "call",  "categoryCd": "call",  "subCategoryCd": "in",  "beforeBalanceAmt": 32.5,  "afterBalanceAmt": 31.5,  "detailId": 0  }, {  "transactionDt": "2015-08-19T08:56:00Z",  "itemTypeCd": "event",  "categoryCd": "adjustment",  "beforeBalanceAmt": 33.5,  "afterBalanceAmt": 32.5,  "detailId": 0  }, {  "transactionDt": "2015-08-16T17:09:00Z",  "itemTypeCd": "event",  "categoryCd": "topup",  "beforeBalanceAmt": 34.5,  "afterBalanceAmt": 33.5,  "detailId": 1  }, {  "transactionDt": "2015-08-13T19:44:00Z",  "itemTypeCd": "call",  "categoryCd": "call",  "subCategoryCd": "out",  "beforeBalanceAmt": 35.5,  "afterBalanceAmt": 34.5,  "detailId": 1  }  ],  "callTransactionList": [  {  "callTypeCd": "longdistance",  "directionCd": "in",  "callingPhoneNum": "4164865688",  "calledPhoneNum": "9056771232",  "voiceMailInd": false,  "durationNum": 1,  "localChargeAmt": 0.5,  "longDistanceChargeAmt": 0.5,  "totalChargeAmt": 1.0,  "rateBreakdownList": [  {  "rateCd": -1,  "rateTxt": "local",  "rateAmt": 0.5,  "rateCountryCd": "CANADA"  }, {  "rateCd": -4,  "rateTxt": "longdistance",  "rateAmt": 0.5,  "rateCountryCd": "CANADA"  }  ],  "cityTxt": "HAMILTON",  "stateTxt": "ON"  }, {  "callTypeCd": "local",  "directionCd": "out",  "callingPhoneNum": "9056771232",  "calledPhoneNum": "9058183443",  "voiceMailInd": false,  "durationNum": 7,  "localChargeAmt": 0.5,  "longDistanceChargeAmt": 0.0,  "totalChargeAmt": 0.5,  "rateBreakdownList": {  "rateCd": -4,  "rateTxt": "longdistance",  "rateAmt": 0.5,  "rateCountryCd": "CANADA"  },  "cityTxt": "HAMILTON",  "stateTxt": "ON"  }  ],  "eventTransactionList": [  {  "eventTypeCd": -5,  "eventTypeTxt": "ADJUSTMENT",  "eventTypeDescTxt": "Account Adjustment",  "confiscatedBalanceAmt": 0,  "airtimeCardId": "",  "featureCd": "",  "featureTxt": "",  "autoTopupAmt": "",  "adjustmentReasonCd": "911Fee",  "adjustmentReasonTxt": "911 Emergency Fee",  "timeZoneOffsetCd": "ET"  }, {  "eventTypeCd": -23,  "eventTypeTxt": "AC\_TOPUP",  "eventTypeDescTxt": "aircard top-up",  "confiscatedBalanceAmt": 0,  "airtimeCardId": "xxxxxxxxxxxx4576",  "featureCd": "",  "featureTxt": "",  "autoTopupAmt": 25,  "adjustmentReasonCd": "",  "adjustmentReasonTxt": "",  "timeZoneOffsetCd": "ET"  }  ]  }  This shows how the detailId for a transactiton points to the corresponding detail depending on whether it’s a call or event. The red line shows the first transaction, which is of type “call”, pointing to the first (row 0) CallTransaction, while the second transaction, of type “event”, has a pointer to the first (row 0) EventTransaction. |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA** | | **SubscriptionEventHistoryService v 1.1**  getCallEventHistory  getSubscriptionEventHistory |  |   Expected average response time: 800 ms |
| **Sample GUI** |  |

## cached-prepaid-call-and-event-history

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/cached-prepaid-call-and-event-history | |
| **Method** | GET |
| **Description** | transactional listing of calls and events which can impact prepaid balance |
| **Input** | **ban** – account number  **sub** – phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | cacheid |  | Mandatory | returned by initial call to prepaid-call-and-event-history | | categoryfilter | “topup”, “adjustment”, “balance”, “feature”, “call”, “in”, “out”, “chargeable”, “free” | Optional | used to filter the list and return a matching subset based on the category. The last four – in/out/chargeable/free – return calls with the matching subcategory. | | start | 10, 20, 30 | no | Starting row of (filtered) result set (1-based list) | | rows | 20, 40 | no | number of rows to return starting with "start" row |   Examples:  **<domain>/prepaidactivity/account/5443434/sub/4165445543/cached-prepaid-call-and-event-history?cacheid=453453400000034&rows=20** – returns first 20 rows of the full history list from cache  **<domain>/prepaidactivity/account/5443434/sub/4165445543/cached-prepaid-call-and-event-history?cacheid=453453400000034&category=adjustment&start=21&rows=40** – returns rows 21 thru 60 of the subset of the history list representing adjustment-related transactions |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IC | Invalid cache |  |  | cache object not found or expired | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  "transactionList": <Transaction> [],  "callTransactionList" : <CallTransaction>[],  "eventTransactionList" : <EventTransaction>[],  "status" : <status>  },  Response will be the same as documented for prepaid-call-and-event-history above, with the exception that a cacheid is not returned. |
| **SLA** | Services/APIs called:  n/a |
| **Sample GUI** |  |

# AccountInfo 2.1/2.2/2.3 REST Service

This service provides operations for retrieving account information.

NOTE: Version 2.3 is the last version to expose the ***account-summaries*** operation. This operation will be removed in future versions. Please see **CustomerInfo** ***summaries*** operation for the equivalent functionality.

## account (GET)

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban} | |
| **Method** | GET |
| **Description** | Get the account information |
| **Input** | **ban** – account number (currently only wireless supported)  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | decorate | "y", "n" | Optional | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated.  **NOTE: decoration not currently implemented** | | language | "en", "fr" | Optional | Language of decoration. Default is "en". | | appid | Indicates which set of resources to decorate with | Optional | Identifies apps with potentially different sets of verbiage; this verbiage would be maintained in RefPDS; e.g. "web-telus", "device-telus", "web-koodo", "device-koodo" | |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | AI-B | Invalid BAN (AIS call failed) | APP10004CMB\_AIS\_0001 | BAN [1234567] Not Found | AIS fails due to invalid BAN – we'll catch this and call it a 400 | | 400 | BAN-I | validation errors: {[BAN is Invalid or Empty]} |  |  |  | | 500 |  | AccountInformationService call failed |  |  | Any other Policy or Service Exception | | 500 |  | general error |  |  | Any caught exception not captured elsewhere | |
| **Output** | {  name : {<name info>},  address : {<address info>},  email: {email address},  billCycleInfo: { <bill\_cycle\_info> },  status : {Status}  }  where **name info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | title | String | Title | Mr | | firstName | String | First Name | John | | lastName | String | Last Name | Smith | | middleInitial | String | Middle Initial | I |   **address info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | status | Status |  |  |   **bill\_cycle\_info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | endDt | date | end date of cycle | e.g. “2013-06-07T08:50:44” | | startDt | date | start date of cycle | e.g. “2013-05-08T08:50:44" | | endDateTxt | String | Localized date | e.g. “23 Juin, 2013” | | startDateTxt | String | Localized date | e.g. “June 23, 2013” | | daysElapsed | integer | number of days into the cycle | 17 | | daysInCycle | integer | number of days in cycle | e.g. 30, 31 | | unitTxt | String | Localized “days” | “days”, “jours” | | billCycleCd | Integer | Bill cycle number | 22 | | captionTxt | string | Caption for cycle info | “Bill cycle”, “Cycle de facturation” | | endsTxt | String | Template for “ends” wording | “ends {0}“, “finit le {0}“ |   Example:  {  “name”:  {  " title" : "Mr",  " firstName" : "John",  " lastName" : "Smith",  " middleInitial" : "I"  },  “address”:  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "poBox" : null,  "rrNum" : null,  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  },  “email”: “sample@telus.com”,  billCycleInfo :  { "endDt":"2013-06-09T08:50:44",  "startDt":"2013-05-10T08:50:44",  "endDateTxt": null,  "startDateTxt": null,  "daysElapsed":27,  "daysInCycle":31,  "unitTxt": null,  "billCycleCd":22,  “captionTxt” : null,  “endsTxt” : null  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## account-summaries

|  |  |
| --- | --- |
| **OPERATION**  account-summaries | |
| **Method** | POST |
| **Version** | 2.1, 2.2, 2.3 **NOTE: In subsequent versions this operation will be removed. Please refer to CustomerInfo service for latest version.** |
| **Description** | Retrieves subscribers, price plan names, and FFH services for the profile |
| **Input** | Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | wirelineCustomerList | 3665647,3665645 | \*Optional | \*Either one of these must be provided  If a wirelessAccountList is passed in, the subscribers within the ban will only be fully decorated if there are less or equal than *maxSubsToDecorateNum* subscribers on the profile (total subscibers on all wirelessAccountList ).  If a subscriberList is passed in, the subscriber objects will always be fully decorated but account object may not be fully decorated. | | wirelessAccountList | 23566656,453657 | \*Optional | | subscriberList | 4165555555,  4165555556 | \*Optional | | maxSubsToDecorateNum | 10 | \*Optional | Will default to 0 if not defined, the total number includes Active and Suspended subs (this number does not affect the subscribers in the subscriberList).  If the number of subs on a profile exceeds *maxSubsToDecorateNum*, none of the subs under a ban will be decorated. | | languageCd | "en" or "fr" | Optional | Language of decoration; default is "en" |   The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "wirelineCustomerList" : "664543,783645”,  "wirelessAccountList" : "5645776,76742345”,  "subscriberList" : "4167778888,4167770000”,  "maxSubsToDecorateNum" : 6,  "languageCd" : "fr"  } |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IC | Invalid Customer | Underlying error code | Underlying error message | CustomerMgmtSvc.Customer.NotFound | | 400 | IB | BAN does not exist | Underlying error code | Underlying error message | Map from domain service:  CMB\_AIS\_0001 | | 400 | IS | Subscriber does not exist | Underlying error code | Underlying error message | Map from domain service:  CMB\_SIS\_0015 | | 400 | AI-OLR | At least one of the 3 input lists is required |  |  |  | | 500 |  | ServiceOrderReferenceService, AccountInformationService, ConsumerCustomerManagementService,  Or SubscriberInformationService failed | Underlying error code | Underlying error message | Any Policy or Service Exception not already captured above | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | Changes added for version 2.1 are highlighted in yellow.  Changes for 2.2 are highlighted in cyan.  Changes for 2.3 are highlighted in magenta.  "wirelessAccountSummaryList" : [<wireless account info>],  "wirelineCustomerSummaryList" : [<wireline customer info>],  "seatGroupingList" : [<seat grouping>],  "standbyModeList" : [<StandbyInfo>],  "status" : <status>  where **wirelessAccountSummaryList** is the list of accounts and where <**wireless** **account info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | name | NameInfo | Name of account holder | NameInfo object | | businessName | string | Name of Business | TELUS | | billingAccountNum | integer | Wireless BAN number | 123556 | | currentBalanceAmt | decimal | Current Balance Due | 56.34 | | accountTypeCd | string | Account Type Code | I | | accountSubTypeCd | string | Account Sub Type Code | R | | mikeInd | boolean | true for Mike account types:  B/1,3,4  C/1,2,3,4,5,6,7,8,J, M,P,R  I/1,2,3 |  | | prepaidInd | boolean | true for acct type / subtype = I/Q |  | | statusCd | string | Account Status | O = Open  C = Closed  N = Cancelled  S = Suspended  T = Tentative | | statusDt | date | Date the status took effect | 2014-03-07T00:00:00-05:00 | | statusActivityReasonCd | string | Reason for the status | SNP,SNP1,SFI,CRC,UDC,UPD = Suspended for non payment  LST = Suspended due to Lost Phone  STL = Suspended due to Stolen Phone  VAD = Vacation Disconnect | | paymentMethodInfo | PaymentMethodInfo | An object containing payment method including cc and bank account |  | | billCycleInfo | BillCycleInfo | An object containing all info relating to billing |  | | subscriberList | SubscriberInfo | A list of objects that contains values pertaining to the sub |  |   where **NameInfo** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | title | String | Title | Mr | | firstName | String | First Name | John | | lastName | String | Last Name | Smith | | middleInitial | String | Middle Initial | I |   where **PaymentMethodInfo**:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | paymentMethodCd | string | Payment method | "creditcard", "bankaccount", "regular" | | creditCardTypeCd | string | Credit Card Type | A – American Express  M – Master Card  V – Visa  O – Other | | creditCardLast4Txt | string | Last 4 digits of registered card | "4577" | | creditCardMonthTxt | string | Expiry month – 2 char | "05" | | creditCardYearTxt | string | Expiry year – 2 char | "03" | | bankCodeTxt | string | Bank institution code | "003" | | bankAccountTxt | string | Bank account number | "3404544" | | bankTransitNumTxt | string | Bank branch (transit) number | "04034" |   where **BillCycleInfo**:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | endDt | date | end date of cycle | e.g. “2013-06-07T00:00:00” |  | | startDt | date | start date of cycle | e.g. “2013-05-08T00:00:00" |  | | endDateTxt | String | Localized date | e.g. “23 juin 2013” | x | | startDateTxt | String | Localized date | e.g. “June 23, 2013” | x | | daysElapsed | integer | number of days into the cycle | 17 |  | | daysInCycle | integer | number of days in cycle | e.g. 30, 31 |  | | billCycleCd | Integer | Bill cycle number | 22 |  |   where **SubscriberInfo**:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | name | NameInfo | Name of subscriber | This will not be populated if the profile has more than maxSubsToDecorateNum subscribers. | | subscriptionId | integer | Subscription ID | 7225303  This will not be populated if the profile has more than maxSubsToDecorateNum subscribers. | | phoneNumberTxt | string | Phone number | 4165555555 | | planDescriptionTxt | string | Price plan name | “Data Flex 35+ Plan”  Will be in the language provided in the input.  This will not be populated if the account has more than maxSubsToDecorateNum subscribers. | | serviceCd | string | Price Plan code | “PCLCHC20”  This will not be populated if the profile has more than maxSubsToDecorateNum subscribers. | | statusCd | string | Status of the subscriber | A – Active  S – Suspended  X – Cancelled – these subs for the most part won't be decorated | | equipmentSerialNumberTxt | string | Serial number of the device associated with the subscriber | 4354654653766 | | provinceCd | string |  | ON, BC | | seatTypeCd | string | seat type – BUSINESS CONNECT ONLY | STRT, MOBL, OFFC, PROF | | seatResourceList | List of SeatResource | BUSINESS CONNECT ONLY |  |   where **SeatResource** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | typeCd | String | resource type | VOIP, TOLLFREE, HSIA, ADDVOIP (additional VOIP) | | resourceNum | String | resource number (phone number or dummy number) | 6045546421 |   where **wirelineCustomerSummaryList** is the list of customers and where <**wireline** **customer info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | customerName | NameInfo | Name of customer | NameInfo object | | customerId | integer | Wireline customer number | 230843922 | | customerTypeCd | string | Account Type Code | I | | customerSubTypeCd | string | Account Sub Type Code | R | | statusCd | string | Status of the custoemr |  | | wirelineAccountSummaryList | WirelineAccountInfo | A list of objects that contains values pertaining to the wireline account |  |   where **wirelineAccountSummaryList** is the list of accounts and where <**wireline** **account info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | accountName | NameInfo | Name of account holder | NameInfo object | | billingAccountNum | integer | Wireline BAN number | 230843922 | | checkDigitTxt | String | check digit for BAN | 3, 4 | | currentBalanceAmt | decimal | Current Balance Due | 56.34 | | statusCd | string | Account Status | O = Open  C = Cancelled | | statusDt | date | Date the status took effect | 2014-03-07T00:00:00-05:00 | | paymentMethodInfo | PaymentMethodInfo | An object containing payment method including cc and bank account |  | | billCycleInfo | BillCycleInfo | An object containing all info relating to billing |  | | productList | ProductInfo | A list of objects that contains values pertaining to the product |  |   where **ProductInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | planDescriptionTxt | string | Description for the plan, only available in English. | "primary price plan - TTV" | | productTypeCd | string | Product code | "TTV" | | statusCd | string | Status of the service | A – Active  S – Suspended | | phoneNumberTxt | string | Phone number for home line | Only populated for home line  "4165555555" | | instanceId | string | Product Catalog ID – used to distinguish between multiple HSIA service instances on a BAN – this can be used as an input to the internet usage operation. | 43074 | | userName | string | For HSIA usage | When there is more than one HSIA service on the BAN, this will contain the username associated with the instance. | | serviceAddress | address (see SeatAddress below) | service address for FFH product – can be used to distinguish multiple instances on a BAN |  |   where **SeatGrouping** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **typical values** | | starterSeatId | String | phone number of this subscriber |  | | subscriptionId | String | subscriptionId of this sub |  | | starterSeatPrimaryInd | boolean | indicates whether this starter seat is the primary one |  | | starterSeatName | NameInfo (see SubscriberInfo above) | Subscriber name (ConsumerName first and last) |  | | starterSeatAddress | SeatAddress (see below) | concatenated address |  | | mobileSeatCount | integer | # of mobile seats associated – cancelled subs not included in total | 5 | | officeSeatCount | integer | # of office seats associated – cancelled subs not included in total | 3 | | professionalSeatCount | integer | # of prof seats associated – cancelled subs not included in total | 4 | | seatList | [SeatInfo] | see below |  |   where **SeatInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | phoneNumberTxt | String | mobile number |  | | subscriptionId | String | subscription id |  | | seatTypeCd | String | type of seat | "MOBL", "OFFC", "PROF" |   where **SeatAddress** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | countryCd | String | Country | "CAN", "USA" |   where **StandbyInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | modeCd | String | indicates type of outage | "wireless", "wireline" | | startDt | timestamp | date/time of start of outage |  | | estimatedCompletionDt | timestamp | ETA of outage completion |  | | lastUpdatedDt | timestamp | time the ETA and/or any messaging was revised |  |   Example:  {  "accountSummariesResponse": {  "wirelessAccountSummaryList": [  {  "name": {  "firstName": "John",  "lastName": "C",  "middleInitial": "D",  "title": "Mr."  },  "businessName" : "TELUS",  "billingAccountNum": 7089796,  "currentBalanceAmt": 57.45,  "accountTypeCd": "I",  "accountSubTypeCd": "R",  "statusCd": "O",  "statusDt": "2014-04-06T00:00:00-04:00",  "statusActivityReasonCd": "CA",  "paymentMethodInfo": {  "paymentMethodCd": "creditcard",  "creditCardTypeCd": "M",  "creditCardLast4Txt": "4567",  "creditCardMonthTxt": "05",  "creditCardYearTxt": "17"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": null,  "startDateTxt": null,  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 22  },  "subscriberList": [  {  "name": {  "firstName": "John",  "lastName": "A",  "title": "Mr."  },  "subscriptionId": 7225303,  "phoneNumberTxt": "4165555555",  "planDescriptionTxt ": "Data Flex 35+ Plan",  "serviceCd": "DFP30",  "statusCd": "A",  "equipmentSerialNumberTxt": "23542545"  }, {  "name": {  "firstName": "John",  "lastName": "B",  "middleInitial": "R",  "title": "Mr."  },  "subscriptionId": 7225304,  "phoneNumberTxt": "4165555556",  "planDescriptionTxt ": "Data Flex 35+ Plan",  "serviceCd": "DFP30",  "statusCd": "A"  }  ]  }, {  "name": {  "firstName": "John",  "lastName": "C",  "title": "Mr."  },  "billingAccountNum": 7089797,  "currentBalanceAmt": 57.45,  "accountTypeCd": "I",  "accountSubTypeCd": "R",  "statusCd": "C",  "statusDt": "2014-04-06T00:00:00-04:00",  "statusActivityReasonCd": "CA",  "paymentMethodInfo": {  "paymentMethodCd": "regular"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 5, 2013",  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 10  },  "subscriberList": [  {  "phoneNumberTxt": "4165555557",  "statusCd": "A"  }, {  "billingAccountNum": 7089797,  "phoneNumberTxt": "4165555558",  "statusCd": "S"  }  ]  }  ],  "wirelineCustomerSummaryList": [  {  "customerId" : 234354,  "name": {  "firstName": "John",  "lastName": "D"  },  "customerTypeCd": "I",  "customerSubTypeCd": "R",  "statusCd": "A",  "wirelineAccountSummaryList": [  {  "name": {  "firstName": "John",  "lastName": "D",  "title": "Mr."  },  "billingAccountNum": 12345,  "currentBalanceAmt": 57.45,  "statusCd": "O",  "statusDt": "2014-04-06T00:00:00-04:00",  "paymentMethodInfo": {  "paymentMethodCd": "bankaccount",  "bankCodeTxt": "003",  "bankAccountTxt": "3404544",  "bankTransitNumTxt": "04034"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 5, 2013",  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 10  },  "productList": [  {  "planDescriptionTxt": "primary price plan - TTV",  "productTypeCd": "TTV",  "statusCd": "A"  }, {  "planDescriptionTxt": "High Speed Turbo 25 (V2)",  "productTypeCd": "HSIC",  "statusCd": "A",  "userName": "jenncyk"  }, {  "planDescriptionTxt": "primary price plan – High Speed",  "productTypeCd": "HSIC",  "statusCd": "A",  "userName": "joecyk"  }, {  "planDescriptionTxt": "primary price plan - single line",  "productTypeCd": "SING",  "phoneNumberTxt": "4033624483",  "statusCd": "A"  }  ]  }  ]  }  ],  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": "2013-06-24T13:45:03-4:00",  "systemErrorCd": 0,  "systemErrorTxt": null  }  }  }  If there are any BusinessConnect accounts in the wirelessAccountList, the response would include something like this:  "seatGroupingList":[  {  "starterSeatId": "7058901397",  "subscriptionId": "26279213",  "starterSeatPrimaryInd":false,  "starterSeatName":{  "firstName": "LOCATIONTWO",  "lastName": "LORI"  },  "starterSeatAddress":{  "streetNum": "300",  "streetName": "CONSILIUM PL",  "streetDirection":null,  "unitTypeCd":null,  "unitNum":null,  "poBox":null,  "rrNum":null,  "cityTxt": "T",  "provinceCd": "ON",  "postalCodeTxt": "M1H3H4",  "addressTypeCd":null,  "countryCd": "CAN",  "stationQualifier":null,  "stationTypeCd":null  },  "mobileSeatCount":1,  "officeSeatCount":1,  "seatList":[  {  "phoneNumberTxt": "4164383488",  "subscriptionId":354343,  "seatType": "office"  },{  "phoneNumberTxt": "4164383489",  "subscriptionId":354773,  "seatType": "mobile"  }  ],  "billingAccountNum": "28509524"  }  ]  If a planned maintenance window (or major unplanned outage which is taking time to resolve) is in place, the response will include the standbyMode section which would look something like this:  "standbyMode" : {  "modeCd": "wireless",  "startDt": "2014-11-23T22:00:00Z",  "estimatedCompletionDt": "2014-11-24T10:00:00Z",  "lastUpdatedDt": "2014-11-24T10:00:00Z"  }, |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 350 ms average; 450 ms peak | | **SubscriberInformationService 3.0**  getSubscriberByPhoneNumber  getSubscriberListByAccountNumber | 275 ms average; 300 ms peak  250 ms average; 350 ms peak | | **ConsumerCustomerManagement 2.2**  getFullCustomerInfo | 110 ms average; 125 ms peak | | **ServiceOrderReferenceService v 4.3** getPricePlan | 175 ms average; 250 ms peak | | **SubscriberProfileConfigurationService** |  | | **BillingAccountDataGridService 1.1** |  | | **SubscriberInformationDataGridService 1.0** |  | | **ConsumerCustomerManagementDataGridService 1.0** |  | | **VoipSupplementaryService 1.0** getOperatorSubscriptionId |  |   Expected average response time: 2000 ms (based on 6 subscribers on the profile)  Expected average response time: 15000 ms (based on 50 subscribers on the profile , and maxSubsToDecorateNum = 50) |
| **Sample GUI** |  |

## ebill-subscription (GET)

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/ebill-subscription | |
| **Method** | GET |
| **Description** | Get account eBill subscription information |
| **Input** | **ban** – account number (currently only wireless supported) |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | BAN-NF | BAN not found |  |  |  | | 400 | BAN-I | validation errors: {[BAN is Invalid or Empty]} |  |  |  | | 500 | SVC1020 | One of the internal TELUS web services is down. Try again later. | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  subscriptionInfo : {<subscription\_info>},  status : {<status>}  }  where **subscription\_info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionStatusInd | boolean | eBill subscription status | true/false | | startDt | date | start date of subscription | e.g. “2013-05-08T08:50:44" | | endDt | date | end date of subscription | e.g. “2013-06-07T08:50:44” |   Example:  {  “subscriptionInfo”:  {  "subscriptionStatusInd" : "true",  "startDt" : "2013-06-09T08:50:44",  "endDt " : "2013-07-09T08:50:44",  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **BillNotificationManagementService v 3.0**  getBillMediumInfo | 2000 ms average; 4000 ms peak |   Expected average response time: 2500 ms |
| **Sample GUI** |  |

## self-serve-subscription (GET)

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/self-serve-subscription | |
| **Method** | GET |
| **Description** | Get account Self Serve subscription information |
| **Input** | **ban** – account number (currently only wireless supported) |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | BAN-NF | BAN not found |  |  |  | | 400 | BAN-I | validation errors: {[BAN is Invalid or Empty]} |  |  |  | | 500 | SVC1020 | One of the internal TELUS web services is down. Try again later. | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  subscriptionInfo : {<subscription\_info>},  status : {<status>}  }  where **subscription\_info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionStatusInd | boolean | Self-Serve subscription status | true/false | | validEmailInd | String | Email validation status | Valid/Invalide | | startDt | date | start date of cycle | e.g. “2013-05-08T08:50:44" | | endDt | date | end date of cycle | e.g. “2013-06-07T08:50:44” |   Example:  {  “subscriptionInfo”:  {  "subscriptionStatusInd" : "true",  "validEmailInd" : " Valid",  "startDt" : "2013-06-09T08:50:44",  "endDt " : "2013-07-09T08:50:44",  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessPermissionService v 3.0**  getRolePermissionListFromObject | 1000 ms average; 3000 ms peak |   Expected average response time: 2000 ms |
| **Sample GUI** |  |

## credit-info (GET)

|  |  |
| --- | --- |
| **OPERATION**  /account/{ban}/credit-info | |
| **Method** | GET |
| **Description** | Get account credit information |
| **Input** | **ban** – account number (currently only wireless supported) |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN (AIS call failed) | APP10004CMB\_AIS\_0001 | BAN [1234567] Not Found | AIS fails due to invalid BAN – we'll catch this and call it a 400 | | 400 |  | validation errors: {[BAN is Invalid or Empty:arg0=]} |  |  |  | | 500 |  | AccountInformationService call failed | <Any downstream error code> | <Any downstream error text> | Any other Policy or Service Exception | | 500 |  | general error | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  creditInfo : {<credit\_info>},  status : {<status>}  }  where **credit\_info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | creditClass | String | Credit class | C,A,.. |   Example:  {  “creditInfo”:  {  "creditClass" : "C",  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByAccountNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

# CustomerInfo REST Service

This service was created as a new home for AccountInfo/account-summaries. The operation has been renamed to simply “summaries”.

CustomerInfo is a general-purpose service which returns information about wireline customers/accounts, wireless accounts and wireless subscribers, such as might be required when rendering an overview page summarizing account and product information. It leverages Datagrid, our high availability/high performance data caching architecture.

|  |
| --- |
| **Notes for Current Consumers of account-summaries**  Please note the following changes:   * **GET** rather than **POST** (hence different request format) * Different domain in URI (**billinginquirymgmt** rather than **selfmgmt**) * Numerics and booleans are now unquoted (e.g. “billCycleCd” : **24** rather than “billCycleCd” : **“24”**) * The response wrapper **accountSummariesResponse** has been stripped away * HTTP Status code is now set to match the Status object statusCd (i.e. it won’t always be 200) |

## summaries

|  |  |
| --- | --- |
| **OPERATION**  Summaries | |
| **Method** | GET |
| **Version** | 1.0, 1.1, 1.2 Changes from 1.0 are highlighted in yellow. |
| **Description** | Retrieves basic information about FFH customers/accounts, wireless accounts, wireless subscribers. BusinessConnect accounts are supported. |
| **Input** | Syntax:  **GET <domain>/v1/cmo/billinginquirymgmt/customerinfo/summaries**  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Description** | **Values** | **Mandatory?** | **Comments** | | wirelinecustomerlist | List of customer IDs | “3665647,3665645” | \*Optional | \*Either one of these must be provided  If a wirelessAccountList is passed in, the subscribers within the ban will only be fully decorated if there are less or equal than *maxSubsToDecorateNum* subscribers on the profile (total subscibers on all wirelessAccountList ).  If a subscriberList is passed in, the subscriber objects will always be fully decorated but account object may not be fully decorated. | | wirelessaccountlist | List of BANs | “23566656,453657” | \*Optional | | subscriberlist | List of phone numbers | “4165555555,4165555556” | \*Optional | | maxdecoratedsubs |  | 10 | \*Optional | Will default to 0 if not defined, the total number includes Active and Suspended subs (this number does not affect the subscribers in the subscriberList).  If the number of subs on a profile exceeds *maxSubsToDecorateNum*, none of the subs under a ban will be decorated. | | maxsubs | max number of subscribers to return per BAN | 100, 1000 | Optional | To improve performance for larger BANs, this will suppress the list of subscribers returned if it exceeds the given threshold. A value of 100, for example, would not return a list subs for a BAN with 1000 subs. If you pass in a BAN with 5 subs and another with 1000 subs, maxsubs=100 would return all 5 subs for the first BAN and none for the second BAN.  IF NOT PROVIDED, all subs will be returned. | | language |  | "en" or "fr" | Optional | Language of decoration; default is "en" |   The parameters will be passed in the URL, e.g.  /v1/cmo/billinginquirymgmt/customerinfo/summaries?wirelessaccountlist=70720769,70724857&wirelinecustomerlist=88338401&maxdecoratedsubs=10&subscriberlist=4167770000&language=en |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | AI-OLR | At least one of the 3 input lists is required |  |  |  | | 500 |  | One of the domain services failed | Underlying error code | Underlying error message | Any Policy or Service Exception not already captured above | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere |   Note:  If valid but non-existent customerID/BAN is passed in, a 400 error will NOT be returned. Instead, a regular 200 is returned with a shell object having statusCd = “X”, e.g.  {"wirelessAccountSummaryList":[{"billingAccountNum":"1476666666","**statusCd":"X"**}],"status":{"statusCd":"200","statusTxt":"OK"}}  If a non-existent subscriber is passed in, a 400 error will NOT be returned; rather no subscriber object will be included in the response for that number, e.g.  {"status": {  "statusCd": "200",  "statusTxt": "OK"  }} |
| **Output** | "wirelessAccountSummaryList" : [<wireless account info>],  "wirelineCustomerSummaryList" : [<wireline customer info>],  "seatGroupingList" : [<seat grouping>],  "standbyModeList" : [<StandbyInfo>],  "status" : <status>  where **wirelessAccountSummaryList** is the list of accounts and where <**wireless** **account info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | name | NameInfo | Name of account holder | NameInfo object | | businessName | string | Name of Business | TELUS | | billingAccountNum | integer | Wireless BAN number | 123556 | | currentBalanceAmt | decimal | Current Balance Due | 56.34 | | accountTypeCd | string | Account Type Code | I | | accountSubTypeCd | string | Account Sub Type Code | R | | mikeInd | boolean | true for Mike account types:  B/1,3,4  C/1,2,3,4,5,6,7,8,J, M,P,R  I/1,2,3 |  | | prepaidInd | boolean | true for acct type / subtype = I/Q |  | | jasperInd | boolean | indicates a corp account administered in Jasper Portal |  | | statusCd | string | Account Status | O = Open  C = Closed  N = Cancelled  S = Suspended  T = Tentative  X = not found | | statusDt | date | Date the status took effect | 2014-03-07T00:00:00-05:00 | | statusActivityReasonCd | string | Reason for the status | SNP,SNP1,SFI,CRC,UDC,UPD = Suspended for non payment  LST = Suspended due to Lost Phone  STL = Suspended due to Stolen Phone  VAD = Vacation Disconnect | | paymentMethodInfo | PaymentMethodInfo | An object containing payment method including cc and bank account |  | | billCycleInfo | BillCycleInfo | An object containing all info relating to billing |  | | activeSubscriberCt | integer | Total number of active subs |  | | suspendedSubscriberCt | integer | Total number of suspended subs |  | | platformCd | String | Available since v1.2  String constant to indicate external system indicator for given account. | ‘MTS’ = Manitoba Telecom System  Other possible values could be present according to source system of account. | | subscriberList | SubscriberInfo | A list of objects that contains values pertaining to the sub |  |   where **NameInfo** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | title | String | Title | Mr | | firstName | String | First Name | John | | lastName | String | Last Name | Smith | | middleInitial | String | Middle Initial | I |   where **PaymentMethodInfo**:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | paymentMethodCd | string | Payment method | "creditcard", "bankaccount", "regular" | | creditCardTypeCd | string | Credit Card Type | A – American Express  M – Master Card  V – Visa  O – Other | | creditCardLast4Txt | string | Last 4 digits of registered card | "4577" | | creditCardMonthTxt | string | Expiry month – 2 char | "05" | | creditCardYearTxt | string | Expiry year – 2 char | "03" | | bankCodeTxt | string | Bank institution code | "003" | | bankAccountTxt | string | Bank account number | "3404544" | | bankTransitNumTxt | string | Bank branch (transit) number | "04034" |   where **BillCycleInfo**:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | endDt | date | end date of cycle | e.g. “2013-06-07T00:00:00” |  | | startDt | date | start date of cycle | e.g. “2013-05-08T00:00:00" |  | | endDateTxt | String | Localized date | e.g. “23 juin 2013” | x | | startDateTxt | String | Localized date | e.g. “June 23, 2013” | x | | daysElapsed | integer | number of days into the cycle | 17 |  | | daysInCycle | integer | number of days in cycle | e.g. 30, 31 |  | | billCycleCd | Integer | Bill cycle number | 22 |  |   where **SubscriberInfo**:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | name | NameInfo | Name of subscriber | This will not be populated if the profile has more than maxSubsToDecorateNum subscribers. | | subscriptionId | integer | Subscription ID | 7225303 | | phoneNumberTxt | string | Phone number | 4165555555 | | planDescriptionTxt | string | Price plan name | “Data Flex 35+ Plan”  Will be in the language provided in the input.  This will not be populated if the account has more than maxSubsToDecorateNum subscribers. | | serviceCd | string | Price Plan code | “PCLCHC20”  This will not be populated if the profile has more than maxSubsToDecorateNum subscribers. | | statusCd | string | Status of the subscriber | A – Active  S – Suspended  X – Cancelled – these subs for the most part won't be decorated | | billingAccountNum | string | BAN of subscriber | 70759470 | | equipmentSerialNumberTxt | string | Serial number of the device associated with the subscriber | 4354654653766 | | provinceCd | string |  | ON, BC | | seatTypeCd | string | seat type – BUSINESS CONNECT ONLY | STRT, MOBL, OFFC, PROF | | seatResourceList | List of SeatResource | BUSINESS CONNECT ONLY |  |   where **SeatResource** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | typeCd | String | resource type | VOIP, TOLLFREE, HSIA, ADDVOIP (additional VOIP) | | resourceNum | String | resource number (phone number or dummy number) | 6045546421 |   where **wirelineCustomerSummaryList** is the list of customers and where <**wireline** **customer info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | customerName | NameInfo | Name of customer | NameInfo object | | customerId | integer | Wireline customer number | 230843922 | | customerTypeCd | string | Account Type Code | I | | customerSubTypeCd | string | Account Sub Type Code | R | | statusCd | string | Status of the custoemr | AC X = not found | | fifaStackInd | boolean | Indicates whether this customer is served out of the FIFA (Falcon) stack | true, false | | teamMemberId | string | Will be populated if customer is a TELUS team member; does not appear otherwise |  | | wirelineAccountSummaryList | WirelineAccountInfo | A list of objects that contains values pertaining to the wireline account |  |   where **wirelineAccountSummaryList** is the list of accounts and where <**wireline** **account info**> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | accountName | NameInfo | Name of account holder | NameInfo object | | billingAccountNum | integer | Wireline BAN number | 230843922 | | checkDigitTxt | String | check digit for BAN | 3, 4 | | currentBalanceAmt | decimal | Current Balance Due | 56.34 | | statusCd | string | Account Status | O = Open  C = Cancelled  X = not found | | statusDt | date | Date the status took effect | 2014-03-07T00:00:00-05:00 | | paymentMethodInfo | PaymentMethodInfo | An object containing payment method including cc and bank account |  | | billCycleInfo | BillCycleInfo | An object containing all info relating to billing |  | | productList | ProductInfo | A list of objects that contains values pertaining to the product |  | | hasFalconProductsInd | boolean | Indicates whether any products int productList is Falcon stack | true/false |   where **ProductInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | planDescriptionTxt | string | Description for the plan, only available in English. | "primary price plan - TTV" | | productTypeCd | string | Product code | "TTV" | | statusCd | string | Status of the service | A – Active  S – Suspended | | phoneNumberTxt | string | Phone number for home line | Only populated for home line  "4165555555" | | instanceId | string | Product Catalog ID – used to distinguish between multiple HSIA service instances on a BAN – this can be used as an input to the internet usage operation. | 43074 | | productBillingId | string | This is another key used to distinguish between multiple HSIA service instances on a BAN. The internet usage operation will require either this value or instanceId. |  | | userName | string | For HSIA usage | When there is more than one HSIA service on the BAN, this will contain the username associated with the instance. | | serviceAddress | address (see SeatAddress below) | service address for FFH product – can be used to distinguish multiple instances on a BAN |  | | serviceStartDt | date yyyy-mm-dd | product instance start date |  | | falconInd | boolean | Product is on Falcon stack | true/false |   where **SeatGrouping** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **typical values** | | starterSeatId | String | phone number of this subscriber |  | | subscriptionId | String | subscriptionId of this sub |  | | starterSeatPrimaryInd | boolean | indicates whether this starter seat is the primary one |  | | starterSeatName | NameInfo (see SubscriberInfo above) | Subscriber name (ConsumerName first and last) |  | | starterSeatAddress | SeatAddress (see below) | concatenated address |  | | mobileSeatCount | integer | # of mobile seats associated – cancelled subs not included in total | 5 | | officeSeatCount | integer | # of office seats associated – cancelled subs not included in total | 3 | | professionalSeatCount | integer | # of prof seats associated – cancelled subs not included in total | 4 | | seatList | [SeatInfo] | see below |  |   where **SeatInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | phoneNumberTxt | String | mobile number |  | | subscriptionId | String | subscription id |  | | seatTypeCd | String | type of seat | "MOBL", "OFFC", "PROF" |   where **SeatAddress** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | countryCd | String | Country | "CAN", "USA" |   where **StandbyInfo** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | modeCd | String | indicates type of outage | "wireless", "wireline" | | startDt | timestamp | date/time of start of outage |  | | estimatedCompletionDt | timestamp | ETA of outage completion |  | | lastUpdatedDt | timestamp | time the ETA and/or any messaging was revised |  |   Example:  {  "wirelessAccountSummaryList": [  {  "name": {  "firstName": "John",  "lastName": "C",  "middleInitial": "D",  "title": "Mr."  },  "businessName" : "TELUS",  "billingAccountNum": 7089796,  "currentBalanceAmt": 57.45,  "accountTypeCd": "I",  "accountSubTypeCd": "R",  "statusCd": "O",  "statusDt": "2014-04-06T00:00:00-04:00",  "statusActivityReasonCd": "CA",  "paymentMethodInfo": {  "paymentMethodCd": "creditcard",  "creditCardTypeCd": "M",  "creditCardLast4Txt": "4567",  "creditCardMonthTxt": "05",  "creditCardYearTxt": "17"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": null,  "startDateTxt": null,  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 22  },  "activeSubscriberCt": 2,  "suspendedSubscriberCt": 0  "subscriberList": [  {  "name": {  "firstName": "John",  "lastName": "A",  "title": "Mr."  },  "subscriptionId": 7225303,  "phoneNumberTxt": "4165555555",  "planDescriptionTxt ": "Data Flex 35+ Plan",  "serviceCd": "DFP30",  "statusCd": "A",  "equipmentSerialNumberTxt": "23542545"  }, {  "name": {  "firstName": "John",  "lastName": "B",  "middleInitial": "R",  "title": "Mr."  },  "subscriptionId": 7225304,  "phoneNumberTxt": "4165555556",  "planDescriptionTxt ": "Data Flex 35+ Plan",  "serviceCd": "DFP30",  "statusCd": "A"  }  ]  }, {  "name": {  "firstName": "John",  "lastName": "C",  "title": "Mr."  },  "billingAccountNum": 7089797,  "currentBalanceAmt": 57.45,  "accountTypeCd": "I",  "accountSubTypeCd": "R",  "statusCd": "C",  "statusDt": "2014-04-06T00:00:00-04:00",  "statusActivityReasonCd": "CA",  "paymentMethodInfo": {  "paymentMethodCd": "regular"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 5, 2013",  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 10  },  "subscriberList": [  {  "billingAccountNum": 7089797,  "phoneNumberTxt": "4165555557",  "statusCd": "A"  }, {  "billingAccountNum": 7089797,  "phoneNumberTxt": "4165555558",  "statusCd": "S"  }  ]  }  ],  "wirelineCustomerSummaryList": [  {  "customerId" : 234354,  "name": {  "firstName": "John",  "lastName": "D"  },  "customerTypeCd": "I",  "customerSubTypeCd": "R",  "statusCd": "A",  "wirelineAccountSummaryList": [  {  "name": {  "firstName": "John",  "lastName": "D",  "title": "Mr."  },  "billingAccountNum": 12345,  "currentBalanceAmt": 57.45,  "statusCd": "O",  "statusDt": "2014-04-06T00:00:00-04:00",  "paymentMethodInfo": {  "paymentMethodCd": "bankaccount",  "bankCodeTxt": "003",  "bankAccountTxt": "3404544",  "bankTransitNumTxt": "04034"  },  "billCycleInfo": {  "endDt": "2013-06-09T08:50:44",  "startDt": "2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 5, 2013",  "daysElapsed": 27,  "daysInCycle": 31,  "billCycleCd": 10  },  "productList": [  {  "planDescriptionTxt": "primary price plan - TTV",  "productTypeCd": "TTV",  "statusCd": "A"  }, {  "planDescriptionTxt": "High Speed Turbo 25 (V2)",  "productTypeCd": "HSIC",  "statusCd": "A",  "userName": "jenncyk"  }, {  "planDescriptionTxt": "primary price plan – High Speed",  "productTypeCd": "HSIC",  "statusCd": "A",  "userName": "joecyk"  }, {  "planDescriptionTxt": "primary price plan - single line",  "productTypeCd": "SING",  "phoneNumberTxt": "4033624483",  "statusCd": "A"  }  ]  }  ]  }  ],  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  }  If there are any BusinessConnect accounts in the wirelessAccountList, the response would include something like this:  "seatGroupingList":[  {  "starterSeatId": "7058901397",  "subscriptionId": "26279213",  "starterSeatPrimaryInd":false,  "starterSeatName":{  "firstName": "LOCATIONTWO",  "lastName": "LORI"  },  "starterSeatAddress":{  "streetNum": "300",  "streetName": "CONSILIUM PL",  "streetDirection":null,  "unitTypeCd":null,  "unitNum":null,  "poBox":null,  "rrNum":null,  "cityTxt": "T",  "provinceCd": "ON",  "postalCodeTxt": "M1H3H4",  "addressTypeCd":null,  "countryCd": "CAN",  "stationQualifier":null,  "stationTypeCd":null  },  "mobileSeatCount":1,  "officeSeatCount":1,  "seatList":[  {  "phoneNumberTxt": "4164383488",  "subscriptionId":354343,  "seatType": "office"  },{  "phoneNumberTxt": "4164383489",  "subscriptionId":354773,  "seatType": "mobile"  }  ],  "billingAccountNum": "28509524"  }  ]  If a planned maintenance window (or major unplanned outage which is taking time to resolve) is in place, the response will include the standbyMode section which would look something like this:  "standbyMode" : {  "modeCd": "wireless",  "startDt": "2014-11-23T22:00:00Z",  "estimatedCompletionDt": "2014-11-24T10:00:00Z",  "lastUpdatedDt": "2014-11-24T10:00:00Z"  }, |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | **ConsumerSubscriberResourceIdentifierMgmtService 1.01** |  | | **BillingAccountDataGridService 1.2** |  | | **SubscriberInformationDataGridService 1.3** |  | | **ConsumerCustomerManagementDataGridService 1.1** |  | | **VoipSupplementaryService 1.0** getOperatorSubscriptionId |  |   Expected average response time: 1000 ms (based on 6 subscribers on the profile)  Expected average response time: 4000 ms (based on 50 subscribers on the profile , and maxSubsToDecorateNum = 50) |

## search (POST)

|  |  |
| --- | --- |
| **OPERATION**  /profile/{uuid}/search | |
| **Method** | POST |
| **Version** | 1.2 |
| **SDF Access** | CII User |
| **Description** | Creates a search result in cache. Search retrieves a list of BANs and subscribers matching/containing a numeric search string, within the wireless BANs and subscribers associated with the given profile |
| **Input** | Syntax:  Via SDF:  **POST <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/search**  Directly (SOA endpoint):  **POST <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/profile/{uuid}/search**  Note that when called via SDF, SDF will inject /profile/{uuid} into the URI, where {uuid} is the profile associated with the Oauth-authenticated login.  Request Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Description** | **Values** | **Mandatory?** | **Comments** | | term | String to search for | “416”, “4165” | Yes | For searchtype=”bansub”, this must consist entirely of numeric characters. | | type | Describes fields to search on | “bansub”, “ban”, “sub” | Optional | Default is “sub”, which indicates that the search will match against subscriber phone number. “ban” will only search against BAN, “sub” will only search against phone number. “bansub” would search both. Future search types could include subscriber name, for example. | | mode | Type of matching | “startswith”, “endswith”, “contains”, “equals” | Optional | Defaults to “contains”. Front end may wish to specify “startswith”, for example, to simplify search and/or to speed it up, as “contains” would typically return more results and be a bit slower. | | sort | Type of sorting | “ban”, “sub”, “name” | Optional | Indicates whether to sort the result set by account number, subscriber number, or name. Default is **sub**. | | cacheminrows | minimum size of result set to cache | 50, 100 | Optional | Default is 20. When the number of matches is greater than or equal to this number, the result set will be cached and a **cacheId** returned (see below). When the number of matches is below this number, the entire result set is returned and no cacheid is returned. Passing in 0 means “do not cache”. | | rows | number of rows, starting from row 1, from result set to return | 10 | Optional | Used in pagination scenarios – represents starting # of rows of full result set to return. Default is 10. |   Example:  **POST <domain>/v1/cmo/billinginquirymgmt/customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search**  {  "term": "4165",  "type": "sub",  "mode": "contains",  "sort": "sub",  "cacheminrows": 20,  "rows": 20  } |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | validation errors: {[Invalid <parameter name> argX=xxxx, etc.] |  |  | All invalid request parameters will be listed e.g. for invalid sort and invalid search type you’d see something like this:  validation errors: {[Invalid sort:arg4=bane], [Invalid search type:arg2=beganwith]} | | 500 | DG | Datagrid call failed | Underlying error code | Underlying error message | Datagrid failure | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "totalMatchesCount" : <integer>,  "resultList" : [<searchResult>],  "cacheId" : <string>  "status" : <status>  where **totalMatchesCount** is the total number of rows matching; **cacheId** is a key to the cached result set to be passed to the subsequent GET in typical pagination scenarios; <**searchResult>** is   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | billingAccountNum | string | account number | “22478665” | | phoneNumberTxt | string | phone number; not populated for BAN matches | “4165765687” | | name | NameInfo (see under /summaries) | Subscriber name (or billing name for an account match) | “John Doe”, “Unit 1” | | subscriberStatusCd | string | subscriber status; not populated for BAN matches | “active”, “suspended” | | resultTypeCd | string | indicates whether match came from phone number (sub) or ban | “ban”, “sub” |   The <searchResult> will not appear in the response unless **start** and **rows** parameters are provided. This is a shortcut to having to make a subsequent call to the GET (see below) to retrieve an initial subset of the search results. In effect it says, “go search according to my criteria and cache the result set server side, and while you’re at it, give me rows a to b so I can display them immediately”. It is the equivalent of making the POST without start/rows, then making a subsequent GET with start/rows.  Example:  {  "totalMatchesCount": 135,  "resultList": [  {  "billingAccountNum": "22478552",  "phoneNumberTxt": "4164535688",  "name": {  "firstName": "John",  "lastName": "B",  "middleInitial": "R",  "title": "Mr."  },  "subscriberStatusCd": "active",  "resultTypeCd": "sub"  }, {  "billingAccountNum": "33045352",  "phoneNumberTxt": "",  "name": {  "firstName": "Jane",  "lastName": "C",  "middleInitial": "Administrator",  "title": "Ms."  },  "subscriberStatusCd": "",  "resultTypeCd": "ban"  } …  ],  "cacheId" : "3a5d-3409b-2343-fc45",  "status": {  "statusCd": 200,  "statusSubCd": "OK"  }  }    Examples:  For the given profile, find all subscribers containing “4165”, and if the result set is 100 or more, cache it:  *customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search*  {  "term": "4165",  "type": "sub",  "mode": "contains",  "sort": "sub",  "cacheminrows": 100,  }  For the given profile, find all subscribers starting with “4165”, and if the result set is 50 or more, cache it, and return the first 20 rows:  *customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search*  {  "term": "4165",  "type": "sub",  "mode": "startswith",  "sort": "sub",  "cacheminrows": 50,  "rows": 20  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | **WirelessPermissionService 3.1** getRolePermissionList |  | | **SubscriberInformationDatagridService 1.6**  searchSubscribers |  | |

## search (GET)

|  |  |
| --- | --- |
| **OPERATION**  /profile/{uuid}/search/{cacheid} | |
| **Method** | GET |
| **Version** | 1.2 |
| **Description** | retrieves from cache a list (or subset) of BANs and subscribers from the previously executed search |
| **Input** | **uuid** - portal profile ID  **cacheid** - reference to cached result set, returned from POST call  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Description** | **Values** | **Mandatory?** | **Comments** | | start | start row of result set to return | 10 | Optional | Used in pagination scenarios – represents starting row of full result set to return; typically used in conjunction with the next parameter **rows**.  Default is 1 (i.e. start with first row). | | rows | number of rows, starting from “start”, from result set to return | 10 | Optional | Used in pagination scenarios – represents starting # of rows of full result set to return, starting with **start**. Default is 10. |   Syntax:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/profile/{uuid}/search/{cacheid}**  Note: the POST (previous operation) will need to be called first to create the search result set and cache it. That operation returns a cacheId which is used as input into this operation.  Example:  Returns rows 1 thru 15 from result set:  */customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search/3a5d-3409b-2343-fc45?start=1&rows=15*  Returns rows 20 thru 29 from result set:  */customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search/3a5d-3409b-2343-fc45?start=20&rows=10*  Returns entire result set:  */customerinfo/profile/019c21f8-5147-4d14-bc0b-cf3997a769b4/search/3a5d-3409b-2343-fc45* |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | ICACHE | Invalid cache ID |  |  | could be caused by expired session | | 400 | IROW | Start row does not exist |  |  | If start parameter (starting row) is size of result set | | 500 |  | general error | Underlying error code | Underlying error message | Any system error | |
| **Output** | "totalMatchesCount" : <integer>,  "resultList" : [<searchResult>],  "status" : <status>  where **totalMatchesCount** is the number of rows in the cached result set (not what’s in the response), and <**searchResult>** is as described in the previous PUT operation.  Example:  {  "totalMatchesCount": 135,  "resultList": [  {  "billingAccountNum": "22478552",  "phoneNumberTxt": "4164535688",  "name": {  "firstName": "John",  "lastName": "B",  "middleInitial": "R",  "title": "Mr."  },  "subscriberStatusCd": "active",  "resultTypeCd": "sub"  }, {  "billingAccountNum": "33045352",  "phoneNumberTxt": "",  "name": {  "firstName": "Jane",  "lastName": "C",  "middleInitial": "Administrator",  "title": "Ms."  },  "subscriberStatusCd": "",  "resultTypeCd": "ban"  } …  ],  "status": {  "statusCd": 200,  "statusSubCd": "OK"  }  } |
| **SLA** | Services/APIs called:  None – this operation retrieves previously obtained results from a Coherence cache. |

## business-hierarchy (Obsolete)

|  |  |
| --- | --- |
| **OPERATION**  business-hierarchy | |
| **Method** | GET |
| **Version** | 1.1 |
| **Auth Level** | OWNER/MANAGER |
| **Description** | Retrieves hierarchical structure of corporate accounts |
| **Input** | Syntax:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/sourceid/{sourceid}/account/{account}/business-hierarchy**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **DataType** | **Mandatory?** | **Comments** | | account | String | Yes | Customer Account Number who may have child accounts in customer hierarchy. | | sourceid | Integer | Yes | Source system id of provided CAN. Account can be sourced from various wireline as well as wireless billing systems | |
| **Output** | Syntax:  {  “hierarchyelementlist”:[<hierarchyelement>\*],  “status”:<status>  }  Response is unnamed JSON object with unordered attributes hierarchyelementlist and status.  If no child accounts are present, hierarchyelementlist will be empty |
| **Status** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**  Syntax:  {  “statusCd”: “<statusCd>”,  “statusSubCd”: “<statusSubCd>”,  “statusTxt”: “<statusTxt>”,  "systemErrorCd": "<systemErrorCd>",  "systemErrorTimeStamp": "<systemErrorTimeStamp>",  "systemErrorTxt": "<systemErrorTxt>"  }  statusCd and statusTxt attributes will always be present in response, other attributes will be populated in case of non success response.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | Validation error text |  |  | Returned when invalid input received; for example alpha numeric source id. | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Hierarchy** | Syntax:  {  "customeraccountnumber":"<string>",  "datasourceid":<int>,  "childcount":<int>,  "billedcustomeraccountnumber":"<string>",  "pilotcustomeraccountnumber":"<string>",  "rcid":"<string>",  "rcidcustomername":"<string>",  "cbucustomername":"<string>",  "customeraccountid":<long>,  "accountstatuscode":"<string>"  }   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | customeraccountnumber | string | Customer accout number of child in hierarchy |  | | datasourceid | Integer | Data source id of customer account | 101, 102 | | childcount | integer | Number of childs of given account.  This attribute is useful to determine if further hierarchy call should be made to get child customer accounts in hierarchy. | 0, 5, 25 | | billedcustomeraccountnumber | string | Billed customer account number. |  | | pilotcustomeraccountnumber | string |  |  | | rcid | string | RCID (child customer ID in ECAT), is a legacy customer ID specifically for a child customer.  An RCID is assigned to each unique TELUS business customer. |  | | rcidcustomername | string | Same as parent account. |  | | cbucid | string | RCIDs are joined hierarchically to a CBUCID. For the majority of our customers,  the CBUCID allows the user to view all of the entities within a customer's organization. |  | | cbucustomername | string | Same as parent account. |  | | customeraccountid | integer | Customer account id in hierarchy database. |  | | accountstatuscode | string |  |  | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | **ClientAPI** Account.getCreditClass  PostpaidConsumerAccount.getCLMSummary |  |   Expected average response time: 1000 ms |

## business-account-summaries

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/ business-account-summaries | |
| **Method** | GET |
| **Version** | 1.1, 1.2 |
| **Auth Level** | CII User |
| **Description** | Retrieves accounts for business accounts, but cross references to ensure all accounts belong to profile |
| **Input** | Syntax:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/profile/{uuid}/ business-account-summaries?canlist={canlist}**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **DataType** | **Mandatory?** | **Comments** | | uuid | String | Yes | User profile UUID as mastered in Client Identity, and inserted into URI by SDF based on user token for CII User protected endpoints. | | canlist | String | No | Customer Account Number list in comma separated format, each with data source id prefixed | | showaddress | String | No | “y” to show address info “n” for no address info Default is “n” |   Example:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/profile/dc3fa905-4e23-4fdc-981f-ea1ca6b07d3d/business-account-summaries?canlist=101-207287198,101-207006302,101-255547164** |
| **Output** | Syntax:  {  “canSummaryList”:[<can summary>\*],  “status”:<status>  }  If no child accounts are present, hierarchyelementlist will be empty  Where **canSummaryList** is the list of accounts to summarize and **<can summary>** is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | billingAccountId | string | Primary key of customer’s billing account number as mastered in Customer Profile DB |  | | billingAccountNum | String | Customers billing account number |  | | dataSourceId | Integer | Master Data Source Id identifying the source billing system for this billing account number | 101, 102 | | regionalCustomerId | string | RCID (child customer ID in ECAT), is a legacy customer ID specifically for a child customer.  An RCID is assigned to each unique TELUS business customer. |  | | rcidCustomerName | string | Name for regional customer. |  | | cbuCustomerId | string | CBUCID, or root level customer identifier.  . |  | | cbuCustomerName | string | CBUCID customer name |  | | customerAccountId | integer | Customer account id in hierarchy database. |  | | accountStatusCd | string | Status of account |  | | accountStatusDt | date | Date of the current status |  | | accountOpenDt | date | Date account was opened |  | | **addressList** |  | See below |  | | **billingNameList** |  |  |  |   Where **addressList** represents:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | addressTxt | string | Rendered address line | “6017 103A ST NW” | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | countryCd | String | Country | "CAN", "USA" |   Where **billingNameList** represents:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | fullName | string | Full rendered name on client account | “CALEDON LABORATORIES LTD” | | nameAssignmentTypeCd | string | Assignment type code for name | “L” | | nameAssignmentSubTypeCd | string | Assignment sub-type code for name | “O” | |
| **Status** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**  Syntax:  {  “statusCd”: “<statusCd>”,  “statusSubCd”: “<statusSubCd>”,  “statusTxt”: “<statusTxt>”,  "systemErrorCd": "<systemErrorCd>",  "systemErrorTimeStamp": "<systemErrorTimeStamp>",  "systemErrorTxt": "<systemErrorTxt>"  }  statusCd and statusTxt attributes will always be present in response, other attributes will be populated in case of non success response.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NP | User not permitted to see any CANs |  |  |  | | 400 | NVC | No valid CANs were passed in |  |  | No billing\_account\_ids were resolved for any CAN in CAN list | | 400 | NPC | No permitted CAN passed in by canlist for this user. |  |  | User has permissions to some CANs, but none that were passed in via canlist. | | 400 | II | Invalid CAN format, all CANs must be formatted srcid-account ie 101-123456789 |  |  | At least one CAN is of invalid format. | | 400 | II | UUID could not be resolved to a profile |  |  | UUID could not be found using call to CIPS | | 400 | II | Status may be Y and N case insensitive values accepted only |  |  | If passing a status, only y or n values are expected. | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Example** | Syntax:  { "canSummaryList": [  {  “billingAccountId”, <int>  "billingAccountNum":"<string>",  "dataSourceId":<int>,  "regionalCustomerId":"<string>",  "rcidCustomerName":"<string>",  "customerId":<long>,  "cbuCustomerName":"<string>",  "customerAccountId":"<int>",  "accountStatusCd":"<string>"  "accountStatusDt":”<date>”  "accountOpenDt":”<date>”  "addressList”: [  {  "addressTxt":"<string>",  "municipalityName":"<string>",  "postalCd":"<string>",  "provinceCd":"<string>",  "streetName":"<string>",  "streetTypeCd":"<string>",  "civicNumberTxt":"<string>"  }  ]  "billingNameList”: [  "fullName":"<string>",  "nameAssignmentSubTypeCd":"<string>",  "nameAssignmentTypeCd":"<string>",  "nameSearchTxt":"<string>",  "organizationName":"<string>"  ]  } ],  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | PermissionDataGridSvc 1.0 *getPermissionList* | 700 ms | | BusinessCustomerManagementDataGridService 1.1 *getAccountIdListByAccountNumberList* *getClientAccountListByCANList* | 1000ms |   Expected average response time: 1000 ms |

## business-account-hierarchy

|  |  |
| --- | --- |
| **OPERATION**  /sourceid/{sourceid}/account/{account}/business-account-hierarchy | |
| **Method** | GET |
| **Version** | 1.1, 1.2 |
| **Auth Level** | OWNER/MANAGER |
| **Description** | Retrieves account for a given business account. Always returns 1 level of depth to hierarchy (current account detaul plus immediate children only) |
| **Input** | Syntax:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/sourceid/{sourceid}/account/{account}/business-hierarchy?status={status}**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **DataType** | **Mandatory?** | **Comments** | | sourceid | String | Yes | Data Master System Source Id for CAN | | account | String | Yes | Customer Account Number | | status | String | No | Status code to filter the list, “A” for active, “C” for closed. Default is all status. |   It would be expected consumers would pass either a billing account id list or a can list.  Example:  **GET <domain>/v1-1/cmo/billinginquirymgmt/customerinfo/sourceid/{sourceid}/account/{account}/business-account-hierarchy?status=a** |
| **Output** | Syntax:  {  “canHierarchyList”:[<hierarchyelement>\*],  “status”:<status>  }  Response is unnamed JSON object with unordered attributes hierarchyelementlist and status.  If no child accounts are present, hierarchyelementlist will be empty  Where **canHierarchyList** is the list of accounts from the hierarchy under the CAN   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | billingAccountId | string | Primary key of customer’s billing account number as mastered in Customer Profile DB | Source system customer’s billing account number | | customerAccountNum | string | Customers billing account number |  | | dataSourceId | integer | Master Data Source Id identifying the source billing system for this billing account number | 101, 102 | | childCt | integer | Number of children account under account passed in.  This attribute is useful to determine if further hierarchy call should be made to get child customer accounts in hierarchy. | 0, 5, 25 | | regionalCustomerId | string | RCID (child customer ID in ECAT), is a legacy customer ID specifically for a child customer.  An RCID is assigned to each unique TELUS business customer. |  | | rcidCustomerName | string | Name for regional customer. |  | | cbuCustomerId | string | CBUCID, or root level customer identifier. |  | | cbuCustomerName | string | CBUCID customer name |  | | customerAccountId | integer | Customer account id in hierarchy database. |  | | accountStatusCd | string | Status of account |  | | accountStatusDt | date | Date of the current status |  | | accountOpenDt | date | Date account was opened |  | | **billingNameList** |  |  |  | | **childAccountList** |  |  |  |   Where **childAccountList** represents:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | billingAccountId | string | Primary key of customer’s billing account number as mastered in Customer Profile DB | Primary key in Customer Profile | | customerAccountNum | string | Customers billing account number | Source system customer’s billing account number | | dataSourceId | integer | Master Data Source Id identifying the source billing system for this billing account number | 101, 102 | | childCt | integer | Number of children account under account passed in.  This attribute is useful to determine if further hierarchy call should be made to get child customer accounts in hierarchy. | 0, 5, 25 | | regionalCustomerId | string | RCID (child customer ID in ECAT), is a legacy customer ID specifically for a child customer.  An RCID is assigned to each unique TELUS business customer. |  | | rcidCustomerName | string | Name for regional customer. |  | | cbuCustomerId | string | CBUCID, or root level customer identifier. |  | | cbuCustomerName | string | CBUCID customer name |  | | customerAccountId | integer | Customer account id in hierarchy database. | Primary key in MBIS | | upRelatedCustomerAccountNum | string |  |  | | billedCustomerAccountNum | string |  |  | | pilotCustomerAccountNum | string |  |  | | accountStatusCd | string | Status of account |  | | accountStatusDt | date | Date of the current status |  | | accountOpenDt | date | Date account was opened |  | | **billingNameList** |  |  |  |   Where **billingNameList** represents:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | fullName | string | Full rendered name on client account | “CALEDON LABORATORIES LTD” | | organizationName | string |  | “CALEDON LABORATORIES” | | nameSearchTxt | string |  | “CALEDON LABORATORIES LTD” | | nameAssignmentTypeCd | string | Assignment type code for name | “L” | | nameAssignmentSubTypeCd | string | Assignment sub-type code for name | “O” | |
| **Status** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**  Syntax:  {  “statusCd”: “<statusCd>”,  “statusSubCd”: “<statusSubCd>”,  “statusTxt”: “<statusTxt>”,  "systemErrorCd": "<systemErrorCd>",  "systemErrorTimeStamp": "<systemErrorTimeStamp>",  "systemErrorTxt": "<systemErrorTxt>"  }  statusCd and statusTxt attributes will always be present in response, other attributes will be populated in case of non success response.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | ICAN | Invalid CAN |  |  |  | | 400 |  | Validation error text |  |  | Returned when invalid input received; for example alpha numeric source id. | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Example** | Syntax:  { " canHierarchyList": [  {  “billingAccountId”, <int>  "billingAccountNum":"<string>",  "dataSourceId":<int>,  "regionalCustomerId":"<string>",  "rcidCustomerName":"<string>",  "customerId":<long>,  "cbuCustomerName":"<string>",  "childCt":<int>,  "accountStatusCd":"<string>"  "accountStatusDt":”<date>”,  "accountOpenDt":”<date>”,  "billingNameList”: [  "fullName":"<string>",  "nameAssignmentSubTypeCd":"<string>",  "nameAssignmentTypeCd":"<string>",  "nameSearchTxt":"<string>",  "organizationName":"<string>"  ]  },  " childAccountList ": [  {  “billingAccountId”, <int>  "billingAccountNum":"<string>",  "dataSourceId":<int>,  "regionalCustomerId":"<string>",  "rcidCustomerName":"<string>",  "customerId":<long>,  "cbuCustomerName":"<string>",  "childCt":<int>,  "accountStatusCd":"<string>"  "accountStatusDt":”<date>”,  "accountOpenDt":”<date>”,  "billingNameList”: [  "fullName":"<string>",  "nameAssignmentSubTypeCd":"<string>",  "nameAssignmentTypeCd":"<string>",  "nameSearchTxt":"<string>",  "organizationName":"<string>"  ]  }  ]  ],  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | BusinessCustomerManagementDataGridService 1.1 *getAccountHierarchyListByCANList* | 1000ms |   Expected average response time: 1000 ms |

## account-status

|  |  |
| --- | --- |
| **OPERATION**  / account/{account}/account-status | |
| **Method** | GET |
| **Version** | 1.1, 1.2 |
| **Auth Level** | OWNER/MANAGER |
| **Description** | Retrieves account specific status attributes |
| **Input** | Syntax:  **GET <domain>/v1-2/cmo/billinginquirymgmt/customerinfo/account/{account}/account-status**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **DataType** | **Mandatory?** | **Comments** | | account | String | Yes | Customer Account Number | |
| **Output** | Response is unnamed JSON object with following attributes   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | delinquentInd | boolean | Boolean property to indicate if account is in delinquent status | true/false | | hotlinedInd | boolean | Boolean property to indicate if account is in hotlined status | true/false | | Status | Status | Status object describing result condition |  | |
| **Status** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**  Syntax:  {  “statusCd”: “<statusCd>”,  “statusSubCd”: “<statusSubCd>”,  “statusTxt”: “<statusTxt>”,  "systemErrorCd": "<systemErrorCd>",  "systemErrorTimeStamp": "<systemErrorTimeStamp>",  "systemErrorTxt": "<systemErrorTxt>"  }  statusCd and statusTxt attributes will always be present in response, other attributes will be populated in case of non success response.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NF | No information found |  |  |  | | 400 | IB | Invalid ban provided |  |  | Validation error returned when ban number is non numeric | | 500 | SF | Downstream service failure |  |  | Error returned when invocation error happened with downstream service calls. | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Example** | Syntax:  {  "delinquentInd": false,  "hotlinedInd": false,  "status": {  "statusCd": "200",  "statusTxt": "OK",  }  } |
| **SLA** | Services/APIs called: |

## linked-accounts (PROPOSED)

|  |  |
| --- | --- |
| **OPERATION**  / profile/{uuid}/linked-accounts | |
| **Method** | GET |
| **Version** | 1.2 |
| **Auth Level** | CII\_USER |
| **Description** | Retrieves accounts linked to a given portal profile |
| **Input** | Syntax:  **GET <domain>/v1-2/cmo/billinginquirymgmt/customerinfo/profile/{uuid}/linked-accounts**   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **DataType** | **Mandatory?** | **Comments** | | uuid | String | Yes | portal profile ID | |
| **Output** | {  “linkedAccountList” : [ <LinkedAccount> list ],  “status” : <Status>  }  where LinkedAccount is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | accountNum | string | BAN | 22478775 | | statusCd | string | account status | “O” = Open  “C” = Closed  “N” = Cancelled  “S” = Suspended  “T” = Tentative  “X” = not found | | systemCd | string | stack – e.g. wireless, ffh | “wls”, “ffh” |   Example:  {  "linkedAccountList": [  {  "accountNum": “34534545”,  "statusCd": "O",  "systemCd": "wls"  }, {  "accountNum": “22475465”,  "statusCd": "S",  "systemCd": "ffh"  }  ],  "status": {  "statusCd": "200",  "statusTxt": "OK"  }  } |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | “UUID could not be resolved to a profile.” |  |  | UUID not found or missing from URI. | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (act)** | | WirelessPermissionService 3.1 *getRolePermissionList* | 250ms | | PermissionDataGridService 1.0  *getRolePermissionList* | 100ms | |  |  |   Expected average response time: 500 ms |

# CreditInfo REST Service

CreditInfo returns information about an account’s credit class, and if applicable, Credit Limit Program details.

## details

|  |  |
| --- | --- |
| **OPERATION**  Details | |
| **Method** | GET |
| **Version** | 1.0 |
| **Auth Level** | OWNER/MANAGER |
| **Description** | Retrieves details about credit class, and where appropriate, Credit Limit Program details. Currently Wireless only. |
| **Input** | Syntax:  **GET <domain>/v1/cmo/billinginquirymgmt/account/{ban}/creditinfo/details**  **ban** – account number |
| **Status Codes** | **As this is a pure REST service, in addition to the status object here, the HTTP Response status code will be set to the value of statusCd in this table.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN |  |  |  | | 500 |  | TelusAPIException |  | exception.getMessage | any general API exception | | 500 |  | general error | Underlying error code | Underlying error message | Any caught exception not captured elsewhere | |
| **Output** | “creditClassCd” : String,  “clpInd” : boolean,  “unpaidBillChargesAmt”: double,  “unpaidUnbilledAmt” : double,  “unpaidAirtimeAmt” : double,  “unpaidDataAmt” : double,  “minimumPaymentAmt” : double,  "status" : <status>   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | creditClassCd | string | single character code indicating credit class, e.g. “B”. | “B”, “L”, “X” | | clpInd | boolean | indicates whether the account is on Credit Limit Program | true, false – credit class L and X are considered CLP | | unpaidBillChargesAmt | double | unpaid bill charges | 134.57 | | unpaidUnbilledAmt | double | unpaid unbilled charges | 56.34 | | unpaidAirtimeAmt | double | unpaid airtime charges | 12.34 | | unpaidDataAmt | double | unpaid data charges | 117.89 | | minimumPaymentAmt | double | minimum payment required to avoid suspension |  | | status | Status |  |  | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Service/operation** | **SLA (actual)** | | **ClientAPI** Account.getCreditClass  PostpaidConsumerAccount.getCLMSummary |  |   Expected average response time: 1000 ms |

# SubscriberInfo REST Service

This service provides operations for retreaving subscriberinformation. Initially this will support wireless only.

## subscriber (GET)

|  |  |
| --- | --- |
| **OPERATION**  /subscriber/{sub} | |
| **Method** | GET |
| **Description** | Get the subscriber information |
| **Input** | **sub** – subscriber phone number |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Subscriber does not exist or has been cancelled | APP20002CMB\_SIS\_0014 | Subscriber not found: Subscriber ID [4165765688] | SIS fails due to invalid sub | | 400 |  | validation errors: {[SubscriberId is Invalid or Empty:arg0=]} |  |  |  | | 500 |  | SubscriberInformationService call failed | <Any downstream error code> | <Any downstream error text> | Any other Policy or Service Exception | | 500 |  | general error | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  name : {<name info>},  address : {<address info>},  email: {email address},  activationDate: { activation\_date }  status : {Status}  }  where **name info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | Title | String | Title | Mr | | firstName | String | First Name | John | | lastName | String | Last Name | Smith | | middleInitial | String | Middle Initial | I |   **address info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | streetNum | String | Street number | 45 | | streetName | String | Street name | King, Jarvis | | unitTypeCd | String | Code for unit type | "APT", "SUITE" | | unitNum | String | Unit number (e.g. apt number) |  | | poBox | String | Post Office Box | 17 | | rrNum | String | Rural Route number |  | | cityTxt | String | City | "Toronto" | | provinceCd | String | Province | "ON", "BC", "QC" | | postalCodeTxt | string | Postal Code | "M5N 2L3" | | status | Status |  |  |   Example:  {  “name”:  {  " title" : "Mr",  " firstName" : "John",  " lastName" : "Smith",  " middleInitial" : "I",  "status" : {….}  },  “address”:  {  "streetNum" : "45",  "streetName" : "King",  "unitTypeCd" : "APT",  "unitNum" : "2707",  "poBox" : null,  "rrNum" : null,  "cityTxt" : "Toronto",  "provinceCd" : "ON",  "postalCodeTxt" : "M5N 2L3"  "status" : {….}  },  “email”: “sample@telus.com”,  “activationDate” : “2013-07-09T00:00:00”  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **SubscribertInformationService v 3.1**  getSubscriberByPhoneNumber | 1000 ms average; 3000 ms peak |   Expected average response time: 1200 ms |
| **Sample GUI** |  |

## self-serve-subscription (GET)

|  |  |
| --- | --- |
| **OPERATION**  /subscriber/{sub}/self-serve-subscription | |
| **Method** | GET |
| **Description** | Get subscriber Self Serve subscription information |
| **Input** | **sub** – subscriber phone number |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Subscriber does not exist or has been cancelled | APP10004CMB\_AIS\_0002 | BAN Not Found |  | | 400 |  | validation errors: {[SubscriberId is Invalid or Empty:arg0=]} |  |  |  | | 500 |  | WirelessPermissionService call failed | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | | 500 |  | ClientIdentityProfileService | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | | 500 |  | AccountInformationService call failed | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | | 500 |  | general error | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  selfServeSubscriptionIinfo : {<self-serve-subscription-info>},  status : {<status>}  }  where **self-serve-subscription-info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | subscriptionStatusInd | boolean | Self-Serve subscription status | true/false | | roleTxt | String | Role Text | Owner | | validEmailInd | String | Email validation status | Valid/Invalide | | profileStatusCd | String | Email validation status | ACTIVE/PENDING | | startDt | date | start date of cycle | e.g. “2013-05-08T08:50:44" | | endDt | date | end date of cycle | e.g. “2013-06-07T08:50:44” | | lastLoginDt | date | Last Login Date | e.g. “2013-06-07T08:50:44” |   Example:  {  “selfServeSubscriptionIinfo”:  {  "subscriptionStatusInd" : "true",  "roleTxt" : " Owner",  "validEmailInd" : " Valid",  "profileStatusCd" : "ACTIVE",  "lastLoginDt” : "2013-06-08T08:50:44",  "startDt" : "2013-06-09T08:50:44",  "endDt " : "2013-07-09T08:50:44",  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **WirelessPermissionService v 3.0**  getRolePermissionListFromObject | 1000 ms average; 3000 ms peak | | **AccountInformationService v 3.1**  getAccountByPhoneNumber | 1000 ms average; 3000 ms peak | | **ClientIdentityProfileService v 2.0**  getProfiles | 3000 ms average; 5000 ms peak |   Expected average response time: 6000 ms |
| **Sample GUI** |  |

## payment-responsible-status (GET)

|  |  |
| --- | --- |
| **OPERATION**  /subscriber/{sub}/payment-responsible-status | |
| **Method** | GET |
| **Description** | Get payment responsible information for subscriber |
| **Input** | **sub** – subscriber phone number |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Subscriber does not exist or has been cancelled | APP10004CMB\_AIS\_0002 | BAN Not Found |  | | 400 |  | validation errors: {[SubscriberId is Invalid or Empty:arg0=]} |  |  |  | | 500 |  | WirelessPermissionService call failed | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | | 500 |  | AccountInformationService call failed | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | | 500 |  | general error | <Any downstream error code> | <Any downstream error text> | Any caught exception not captured elsewhere | |
| **Output** | {  paymentResponsibleInfo: {< payment-responsible-info>},  status : {<status>}  }  where **payment-responsible-info** is as follows:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | paymentResponsibleInd | boolean | Payment responsible | true/false | | paymentResponsibleSubscriber | String | Payment responsible subscriber phone number | 4168973212 |   Example:  {  “paymentResponsibleInfo”:  {  " paymentResponsibleInd" : "true",  “paymentResponsibleSubscriber” : “4168973212”,  "status" : {….}  } |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountInformationService v 3.1**  getAccountByPhoneNumber | 1000 ms average; 3000 ms peak | | **WirelessPermissionService v 3.0**  getRolePermissionListFromObject | 1000 ms average; 3000 ms peak |   Expected average response time: 2000 ms |
| **Sample GUI** |  |

# InvoiceInfo REST Service

This service supports bill presentment operations.

## bill-summary

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-summary  sourceid/{sourceid}/account/{ban}/bill-summary | |
| **Method** | GET |
| **Description** | Return the latest bill summary for the given BAN/CAN with service level charges |
| **Input** | **ban** – account number (currently only wireless postpaid, compass (wireline consumer and single line business) supported  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Type** | **Comments** | | system | "wls", "ffh" | Y |  | Billing system – default is wireless (wls) | | units | Number, 1-10 | Y | WLS | By default, the service does not return service level information. This parameter lets the consumer the control the number of services returned. | | pending | “y”, “n” | Y | WLS | By default, it is “n”, it returns latest released bill. If it is “y”, the result will show pending bill. | | showLineItem | “y”, “n” | Y |  | By default, it returns no line items for service level sections. If this is “y”, then the service level line items will be shown. | | decorate | “y”, “n” | Y |  | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="n"), response will not be decorated. | | showcollectionresource | “y”, “n” | Y | WLS | Indicates whether it needs to show resource list associated with business connect services. By default, it returns no resources | | billIndex | 0, -1, -2, -3 | Y |  | Bill index values to allow fetching previous bills  0 is default, corresponds to latest bill. | | sourceid | 101, 102, 1001, 130, 119 | Y |  | 101 -> CORE AB  102 -> CORE BC  1001 -> Enabler  130 -> KB  119 -> TQ COR |   Examples:  account/2343465/bill-summary  account/2343465/bill-summary?units=5&decorate=n |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given BAN |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billSummaryResponse" : {  “billSummary”: {  "billDocumentId": integer,  "billLanguageCd ": String,  “billDt”: date,  “billStartDt”: date,  “billEndDt”: date,  “nextBillCycleStartDt”: date,  “nextBillCycleEndDt”: date,  "billingAccountNum": integer,  "brandId": integer,  "corporateInd": String,  "convergedBillInd": String,  “paymentDueDt”: date,  “additionalUsageInd”: String,  “totalSavingsAmt”: decimal,  “balanceForwardAmt”: decimal,  “lastTotalInvoiceAmt”: decimal,  “totalInvoiceAmt”: decimal,  “totalTaxAmt”: decimal,  “totalDueAmt”: decimal,  “bundleAmt”: decimal,  “bundleAmtList”: [<chargeItem>],  “OtherChargeTotalAmt”: decimal,  “OtherChargeItemList”: [<chargeItem>],  “serviceCollectionGroupSummaryList”: [  {  groupId: integer,  descriptionTxt: string,  preTaxChargeAmt: decimal,  partialChargeAmt: decimal,  monthlyChargeAmt: decimal,  additionalChargeAmt: decimal,  addOnAmt: decimal,  taxAmt: decimal,  totalChargeAmt: decimal,  savingAmt: decimal,  collectionTypeSummaryList: [<collectionTypeSummary>]  }  ]  “serviceTypeSummaryList”: [  <serviceTypeSummary>  ],  "serviceList": [  {  "serviceInfo": {  billServiceInstanceId: integer,  serviceTypeCd: string,  primaryServiceResourceId: string,  serviceInstanceSeqNum: integer,  serviceUserName: string,  serviceCollectionTypeCd: string,  serviceCollectionGroupId: integer,  serviceCollectionResourceList: [<serviceCollectionResource>]  },  "serviceCharge":  {  “preTaxChargeAmt”: Decimal,  “monthlyChargeAmt”: Decimal,  “monthlyChargeItemList”: [<chargeItem>],  “partialChargeAmt”: Decimal,  “partialChargeItemList”: [<chargeItem>],  “addOnAmt”: Decimal,  “addOnList”: [<chargeItem>],  “taxAmt”: Decimal,  “totalChargeAmt”: Decimal,  “savingAmt”: Decimal  },  “usageChargeTotalAmt”: decimal,  “usageChargeList”: [<usageCharge>],  “taxItemList”: [<taxDisplay>]  }  ],  “billTax”: {  "taxTypeSummaryList": [<taxDisplay>],  "serviceTaxSummaryList": [<taxDisplay>],  "acctTaxItemList": [<taxDisplay>]  }  },  "status” : <status>  }   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Attr**  **Type** | **Opt** | **Possible/typical values** | **Decoration** | | billDocumentId | integer | Bill document ID |  |  | “1234556” |  | | billLanguageCd | String | Bill language |  |  | “EN” |  | | billingAccountNum | integer | Account number |  |  | “1234556” |  | | brandId | integer | Brand ID | WLS |  | “1” |  | | corporateInd | String | Corporate Indicator | WLS |  | “true” |  | | billDt | date | Bill date |  |  | “2013-10-20” |  | | billStartDt | date | Bill start date |  |  | “2013-10-20” |  | | billEndDt | date | Bill end date |  |  | “2013-11-20” |  | | nextBillCycleStartDt | date | Next bill cycle start date |  |  | “2013-11-20” |  | | nextBillCycleEndDt | date | Next bill cycle end date |  |  | “2013-11-20” |  | | convergedBillInd | String | Converged Bill indicator | WLS |  | “true” |  | | paymentDueDt | date | Payment Due date |  |  | “2013-06-07-04:00” |  | | additionalUsageInd | String | Show whether any service has additional usage |  |  | “true”, “false” |  | | taxTypeSummaryList | List | Show the total tax breakdown on bill level | WLS | Y | List |  | | serviceTaxSummaryList | List | Show the total tax breakdown on the service summary level | WLS | Y | List |  | | acctTaxItemList | List | Show the total tax breakdown on the account level charges | WLS | Y | List |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | totalSavingsAmt | decimal | Saving Amount | Y | WLS | “120.00” |  | | balanceForwardAmt | decimal | Balance forward from last bill |  |  | “120.00” |  | | totalInvoiceAmt | decimal | Total charge Amount with tax included |  |  | “120.00” |  | | lastTotalInvoiceAmt | decimal | Total charge Amount for last bill | Y |  | “120.00” |  | | totalTaxAmt | decimal | Total tax Amount |  |  | “120.00” |  | | totalDueAmt | decimal | Total Amount due with balance forward included |  |  | “120.00” |  | | bundleAmt | decimal | Total Amount for bundle services | Y | FFH | “120.00” |  | | bundleAmtList | List | List of cross bundle charges | Y | FFH | <chargeItem> |  | | OtherChargeTotalAmt | Decimal | Other charges and credit summary amount | Y |  | “120.00” |  |   Where <serviceTypeSummary> is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | **Decoration** | | serviceTypeSummaryList | List | List of serviceTypeSummary | Y | [<serviceTypeSummary] |  | | serviceTypeCd | String | Type of service with possible values, “MOBILE” (wireless), “INTERNET” (High speed internet), “TV” (Telus TV), “LANDLINE” (home phone), and “BUSINESS\_SERVICE” (business service), “BUSINESS\_CONNECT” (Business Connect Seat) |  | “MOBILE”, “INTERNET”, “TV”, “LANDLINE”, “BUSINESS\_CONNECT”, “BUSINESS\_SERVICE” |  | | totalPreTaxChargeAmt | decimal | Total Pre-tax charges for this particular service type |  | “120.00” |  | | totalServiceCount | integer | Total number of services for this particular service type |  | 3 |  |   Where “serviceCollectionGroupSummaryList” is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr Type** | **Possible/typical values** | | groupId | Integer | Service collection group identifier |  | WLS | “35424242” | | descriptionTxt | string | Group total Pre-tax charges |  | WLS | “300 Consilium Place..” | | preTaxChargeAmt | decimal | Group pre-tax total service Amount |  | WLS | “120.00” | | partialChargeAmt | decimal | Group partial charges | Y | WLS | “120.00” | | monthlyChargeAmt | decimal | Group monthly/Regular Charges |  | WLS | “120.00” | | additionalChargeAmt | Decimal | Group additional charge amount | Y | WLS | “120.00” | | addOnAmt | decimal | Group add-on Amount | Y | WLS | “120.00” | | taxAmt | decimal | Group tax on charges |  | WLS | “120.00” | | totalChargeAmt | decimal | Group total charges |  | WLS | “120.00” | | savingAmt | decimal | Group Saving | Y | WLS | “120.00” | | collectionTypeSummaryList | List | Summary list containing collection types |  | WLS | <collectionTypeSummary> |   Where <collectionTypeSummary> is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | | serviceCollectionTypeCd | String | Business Connect service type |  | WLS | “STARTER”, “OFFICE”,“MOBILE”, “PROFESSIONAL” | | totalServiceCount | integer | Total number of services for this particular collection type |  |  | “5” |   Where <chargeItem> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | chargeItemDescriptionTxt | decimal | Description |  |  | “Paper bill Fee” |  | | chargeItemAmt | decimal | Balance forward from last bill |  |  | “120.00” |  | | chargeItemAmtTxt | String | Text representation of the charge item amount. i.e. "FREE" , "0" , “B“ (which means suppress amount) , or the exact value as chargeItemAmt |  | FFH | “120.00”, “0”, “Free”, “B” |  | | servicePackageGroupCd | String | Business Connect attribute showing the resource type associated with the charge | Y | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | bundleChargeItemSubList | List | A list of the sub items <chargeItem> as bundle offer | Y | FFH | <chargeItem> | N |   Where “serviceInfo” is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | | billServiceInstanceId | integer | Bill Service Instance ID |  |  | “1” | | serviceTypeCd | String | Type of service with possible values, “MOBILE” (wireless), “INTERNET” (High speed internet), “TV” (Telus TV), “LANDLINE” (home phone), and “BUSINESS\_SERVICE” (business service), “BUSINESS\_CONNECT” (Business Connect) mapped to the service instance class code in billing ODS |  |  | “MOBILE”, “INTERNET”, “TV”, “LANDLINE”, “BUSINESS\_SERVICE”, “BUSINESS\_CONNECT” | | primaryServiceResourceId | String | Cell number for wireless service | Y |  | “4169999999” | | serviceInstanceSeqNum | Integer | Service Instance Sequence number | Y | WLS | “0” | | serviceUserName | String | Service user name | Y | WLS | “DARCY MCGUIRE” | | serviceCollectionTypeCd | String | Business Connect service type | Y | WLS | “STARTER”, “OFFICE”,“MOBILE”, “PROFESSIONAL” | | serviceCollectionGroupId | Integer | Business Connect group location Id | Y | WLS | “54594” | | serviceCollectionResourceList | List | List of service collection resources | Y | WLS | [<serviceCollectionResource>] |   Where <serviceCollectionResource> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | resourceId | Integer | Resource Id |  | WLS | “4542123” |  | | resourceTypeCd | String | Resource type code |  | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | displayTxt | String | Invoice display text |  | WLS | “Toll Free number” |  | | resourceValueTxt | String | The identifier for the resource, i.e. phone number |  | WLS | “18665452365” |  |   Where <serviceCharge> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | preTaxChargeAmt | decimal | Pre-tax total service Amount |  |  | “120.00” |  | | partialChargeAmt | decimal | Partial charges | Y |  | “120.00” |  | | monthlyChargeAmt | decimal | Monthly/Regular Charges |  |  | “120.00” |  | | addOnAmt | decimal | Add-on Amount | Y | WLS | “120.00” |  | | taxAmt | decimal | Tax on charges |  | WLS | “120.00” |  | | totalChargeAmt | decimal | Total Charges tax included |  | WLS | “120.00” |  | | savingAmt | decimal | Saving Amount | Y | WLS | “120.00” |  | | usageChargeTotalAmt | decimal | Total usage summary charge | Y |  | “120.00” |  | | monthlyChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  | | partialChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  | | addOnList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  |   Where <usageCharge> is:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | usageSummaryTypeCd | String | Type of usage summary with possible values, “AIRTIME”, “DATA”, “OTHER”, “TV”, “INTERNET”, “LANDLINE”, mapped to the usage summary type code in billing ODS | “AIRTIME”, “DATA”, “OTHER”, “TV”, “INTERNET”, “LANDLINE”, “VOIP” | | usageSummaryTxt | String | Usage summary description text | “Airtime”, “Data”, “Other”, “TV”, “INTERNET”, “LANDLINE” | | usageSummaryChargeAmt | decimal | Usage summary charges | “120.00” |   Where <taxDisplay> is   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | displayText | String | Tax Display text | “GST / HST”, “GST”, “PST” | | taxAmt | decimal | Tax amount | “12.34” | | taxTypeCd | String | Tax type code | “GHST”, “HST”, “PST” | | taxAuthCd | String | Tax authorization code | “GOV” |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "billSummaryResponse": {  "billSummary": {  "billDocumentId": 100001,  "billLanguageCd": "EN",  "billingAccountNum": 7563242,  "brandId": 4,  "corporateInd": "n",  "additionalUsageInd": "y",  "convergedBillInd": "n",  "paymentDueDt": "2013-06-07-04:00",  "totalSavingsAmt": 12.5,  "balanceForwardAmt": 50,  "lastTotalInvoiceAmt": 20,  "totalInvoiceAmt": 200,  "totalTaxAmt": 200,  "totalDueAmt": 150,  "bundleAmt": 30,  "OtherChargeTotalAmt": 5,  "OtherChargeItemList": [  {  "chargeItemDescriptionTxt": "Late payment charge",  "chargeItemAmt": 3  },  {  "chargeItemDescriptionTxt": "Summary Paper Bill Fee",  "chargeItemAmt": 2  }  ],  "serviceTypeSummary": [  {  "serviceTypeCd": "MOBILE",  "totalPreTaxChargeAmt": 160,  "totalServiceCount": 50  },  {  "serviceTypeCd": "MOBILE",  "totalPreTaxChargeAmt": 30,  "totalServiceCount": 20  }  ],  "serviceList": [  {  "serviceInfo": {  "billServiceInstanceId": 13423423,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992929",  "serviceInstanceSeqNum": 10,  "serviceUserName": "Kevin Lo"  },  "serviceCharges": {  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "partialChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "monthlyChargeAmt": 100,  "monthlyChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  },  "usageChargeTotalAmt": 45.5,  "usageChargeList": [  {  "usageSummaryTypeCd": "DATA",  "usageSummaryTxt": "Data",  "usageSummaryChargeAmt": 25.5  },  {  "usageSummaryTypeCd": "VOICE",  "usageSummaryTxt": "Airtime",  "usageSummaryChargeAmt": 20  },  {  "usageSummaryTypeCd": "Others",  "usageSummaryTxt": "OTHER",  "usageSummaryChargeAmt": 20  }  ]  },  {  "serviceInfo": {  "billServiceInstanceId": 13423410,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992119",  "serviceInstanceSeqNum": 11,  "serviceUserName": "Zhong Ke"  },  "serviceCharges": {  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  },  "usageChargeTotalAmt": 65.5,  "usageChargeList": [  {  "usageSummaryTypeCd": "DATA",  "usageSummaryTxt": "DATA",  "usageSummaryChargeAmt": 25.5  },  {  "usageSummaryTypeCd": "VOICE",  "usageSummaryTxt": "Airtime",  "usageSummaryChargeAmt": 20  }  ]  },  {  "serviceInfo": {  "billServiceInstanceId": 13423499,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992519",  "serviceInstanceSeqNum": 12,  "serviceUserName": "Jim Cai"  },  "serviceCharges": {  "preTaxChargeAmt": 100,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 300,  "savingAmt": 10  },  "usageChargeTotalAmt": 0  }  ]  },  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } | billCycleInfo :  { "endDt":"2013-06-09T08:50:44",  "startDt":"2013-05-10T08:50:44",  "endDateTxt": "June 9, 2013",  "startDateTxt": "May 10, 2013",  "daysElapsed":27,  "daysInCycle":31,  "billCycleCd":22,  “captionTxt” : "Bill cycle",  “endsTxt” : "Ends {0}"  },  Status :  { “statusCd” : 200,  “statusTxt” : “OK”,  “statusDescriptionTxt” : null,  "systemErrorTimeStamp": null,  “systemErrorCd : 0,  “systemErrorTxt” : null  } | |
| **SLA** | Expected average response time: 2000 ms, 4000 ms peak |

## service-summary

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/service-summary  sourceid/{sourceid}/account/{ban}/service-summary | |
| **Method** | GET |
| **Description** | Return the latest list of services summary for the given BAN/CAN with optional search criteria |
| **Input** | **ban** – account number (currently only wireless postpaid, compass (wireline consumer and single line business) supported  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Type** | **Comments** | | system | "wls", "ffh" | Y |  | Billing system – default is wireless (wls) | | username | “Kevin” | Y | WLS | Search by name associated with the service | | phonenumber | 4169999999 | Y | WLS | Search by phone number | | overagetype | “AIRTIME”, “DATA”, “OTHER” | Y | WLS | Search by overage type | | start | 1 | Y | WLS | By default, it returns first to fifth service subscriptions with respect to any search criteria. Once this is set, it returns the service list specified by the start and end index | | end | 5 | Y | WLS | By default, it returns first to fifth service subscriptions with respect to any search criteria. Once this is set, it returns the service list specified by the start and end index | | pending | “y”,”n” | Y | WLS | By default, it is “n”, it returns latest released bill. If it is “y”, the result will show pending bill. | | showLineItem | “y”, “n” | Y |  | By default, it returns no line items for service level sections. If this is “y”, then the service level line items will be shown. | | showtotal | “y”, “n” | Y | WLS | By default, it returns no total number of search results. | | collectiongrouptype | “STARTER”, “MOBILE”, “OFFICE”, “PROFESSIONAL” | Y | WLS | Search by Business Connect service type | | decorate | “y”, “n” | Y |  | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated. | | showcollectionresource | “y”, “n” | Y | WLS | Indicates whether it needs to show resource list associated with business connect services. By default, it returns no resources | | billIndex | 0, -1, -2, -3 | Y |  | Bill index values to allow fetching previous bills. 0 is default, corresponds to latest bill. | | sourceid | 101, 102, 1001, 130, 119 | Y |  | 101 -> CORE AB  102 -> CORE BC  1001 -> Enabler  130 -> KB  119 -> TQ COR |   Examples:  account/2343465/service-summary  account/2343465/service-summary?phonenumber=4169999999&username=Kevin&decorate=n  account/2343465/service-summary?phonenumber=4169999999&username=Kevin&start=1&end=5&decorate=n |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given parameters |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "serviceSummaryResponse" : {  "serviceList": [  {  "serviceInfo": {  billServiceInstanceId: integer,  serviceTypeCd: string,  primaryServiceResourceId: string,  serviceInstanceSeqNum: integer,  serviceUserName: string,  serviceCollectionTypeCd: string,  serviceCollectionGroupId: integer,  serviceCollectionResourceList: [<serviceCollectionResource>]  },  "serviceCharge":  {  “preTaxChargeAmt”: Decimal,  “monthlyChargeAmt”: Decimal,  “monthlyChargeItemList”: [<chargeItem>],  “partialChargeAmt”: Decimal,  “partialChargeItemList”: [<chargeItem>],  “addOnAmt”: Decimal,  “addOnList”: [<chargeItem>],  “taxAmt”: Decimal,  “totalChargeAmt”: Decimal,  “savingAmt”: Decimal  },  “usageChargeTotalAmt”: decimal,  “usageChargeList”: [<usageCharge>],  “taxItemList”: [<taxDisplay>]  }  ],  “totalCount”: integer,  "serviceTaxSummaryList": [<taxDisplay>],  "status” : <status>  }   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | | totalCount | integer | Total number of search results, will be shown if showtotal=y | Y | WLS | “6” | | serviceTaxSummaryList | List | Show the total tax breakdown on the service summary level | Y | WLS | List |   Where “serviceInfo” is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | | billServiceInstanceId | integer | Bill Service Instance ID |  |  | “1” | | serviceTypeCd | String | Type of service with possible values, “MOBILE” (wireless), “INTERNET” (High speed internet), “TV” (Telus TV), “LANDLINE” (home phone), and “BUSINESS\_SERVICE” (business service), “BUSINESS\_CONNECT” (Business Connect Seat) |  |  | “MOBILE”, “INTERNET”, “TV”, “LANDLINE”, “BUSINESS\_CONNECT”, “BUSINESS\_SERVICE” | | primaryServiceResourceId | String | Cell number for wireless service | Y |  | “4169999999” | | serviceInstanceSeqNum | Integer | Service Instance Sequence number | Y | WLS | “0” | | serviceUserName | String | Service user name | Y | WLS | “DARCY MCGUIRE” | | serviceCollectionTypeCd | String | Business Connect service type | Y | WLS | “STARTER”, “OFFICE”,“MOBILE”, “PROFESSIONAL” | | serviceCollectionGroupId | Integer | Business Connect group location Id | Y | WLS | “54594” | | serviceCollectionResourceList | List | List of service collection resources | Y | WLS | [<serviceCollectionResource>] |   Where <serviceCollectionResource> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | resourceId | Integer | Resource Id |  | WLS | “4542123” |  | | resourceTypeCd | String | Resource type code |  | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | displayTxt | String | Invoice display text |  | WLS | “Toll Free number” |  | | resourceValueTxt | String | The identifier for the resource, i.e. phone number |  | WLS | “18665452365” |  |   Where “serviceCharge” is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Type** | **Possible/typical values** | **Decoration** | | preTaxChargeAmt | decimal | Pre-tax total service Amount |  |  | “120.00” |  | | partialChargeAmt | decimal | Partial charges | Y |  | “120.00” |  | | monthlyChargeAmt | decimal | Monthly/Regular Charges |  |  | “120.00” |  | | addOnAmt | decimal | Add-on Amount | Y | WLS | “120.00” |  | | taxAmt | decimal | Tax on charges |  | WLS | “120.00” |  | | totalChargeAmt | decimal | Total Charges tax included |  | WLS | “120.00” |  | | savingAmt | decimal | Saving Amount | Y | WLS | “120.00” |  | | usageChargeTotalAmt | decimal | Total usage summary charge | Y |  | “120.00” |  | | monthlyChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  | | partialChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  | | addOnList | List | A list of detail charge items, will be shown based on input showLineItem | Y |  | <chargeItem> |  |   Where <chargeItem> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | chargeItemDescriptionTxt | decimal | Description |  |  | “Paper bill Fee” |  | | chargeItemAmt | decimal | Balance forward from last bill |  |  | “120.00” |  | | chargeItemAmtTxt | String | Text representation of the charge item amount. i.e. "FREE" , "0" , “B“ (which means suppress amount) , or the exact value as chargeItemAmt |  | FFH | “120.00”, “0”, “Free”, “B” |  | | servicePackageGroupCd | String | Business Connect attribute showing the resource type associated with the charge | Y | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | bundleChargeItemSubList | List | A list of the sub items <chargeItem> as bundle offer | Y | FFH | <chargeItem> | N |   Where <usageCharge> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | usageSummaryTypeCd | String | Type of usage summary with possible values, “AIRTIME”, “DATA”, “OTHER”, “ON-DEMAND”, “PAY-PER-VIEW”, mapped to the usage summary type code in billing ODS | “AIRTIME”, “DATA”, “OTHER”, “TV”, “INTERNET”, “LANDLINE”, “VOIP” |  | | usageSummaryTxt | String | Usage summary description text | “Airtime”, “Data”, “Other”, “TV”, “INTERNET”, “LANDLINE” | X | | usageSummaryChargeAmt | decimal | Usage summary charges | “120.00” |  |   Where <taxDisplay> is   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | displayText | String | Tax Display text | “GST / HST”, “GST”, “PST” | | taxAmt | decimal | Tax amount | “12.34” | | taxTypeCd | String | Tax type code | “GHST”, “HST”, “PST” | | taxAuthCd | String | Tax authorization code | “GOV” |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "serviceSummaryResponse": {  "serviceList": [  {  "serviceInfo": {  "billServiceInstanceId": 13423423,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992929",  "serviceInstanceSeqNum": 10,  "serviceUserName": "Kevin Lo"  },  "serviceCharges": {  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "partialChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "monthlyChargeAmt": 100,  "monthlyChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  },  "usageChargeTotalAmt": 45.5,  "usageChargeList": [  {  "usageSummaryTypeCd": "DATA",  "usageSummaryTxt": "Data",  "usageSummaryChargeAmt": 25.5  },  {  "usageSummaryTypeCd": "VOICE",  "usageSummaryTxt": "Airtime",  "usageSummaryChargeAmt": 20  },  {  "usageSummaryTypeCd": "OTHER",  "usageSummaryTxt": "Other",  "usageSummaryChargeAmt": 20  }  ]  },  {  "serviceInfo": {  "billServiceInstanceId": 13423410,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992119",  "serviceInstanceSeqNum": 11,  "serviceUserName": "Zhong Ke"  },  "serviceCharges": {  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "partialChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "monthlyChargeAmt": 100,  "monthlyChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  },  "usageChargeTotalAmt": 65.5,  "usageChargeList": [  {  "usageSummaryTypeCd": "DATA",  "usageSummaryTxt": "Data",  "usageSummaryChargeAmt": 25.5  },  {  "usageSummaryTypeCd": "VOICE",  "usageSummaryTxt": "Airtime",  "usageSummaryChargeAmt": 20  }  ]  },  {  "serviceInfo": {  "billServiceInstanceId": 13423499,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992519",  "serviceInstanceSeqNum": 12,  "serviceUserName": "Jim Cai"  },  "serviceCharges": {  "preTaxChargeAmt": 100,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 300,  "savingAmt": 10  },  "usageChargeTotalAmt": 0  }  ],  “totalCount”: 2  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 2000 ms, 4000 ms peak |

## subscriber-summary

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/sub/{sub}/subscriber-summary  sourceid/{sourceid}/account/{ban}/sub/{sub}/subscriber-summary | |
| **Method** | GET |
| **Description** | **(Wirelss only)** Return the latest subscriber level bill summary, **this is only available for wireless bills (WLS)** |
| **Input** | **ban** – account number  **sub** – subscriber phone number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Comments** | | pending | “y”,”n” | Y | By default, it is “n”, it returns latest released bill. If it is “y”, the result will show pending bill. | | showLineItem | “y”, “n” | Y | By default, it returns no line items for service level sections. If this is “y”, then the service level line items will be shown. | | decorate | “y”, “n” | Y | Indicates that response is to include standardized verbiage and localized dates/numbers. If not provided (or value="N"), response will not be decorated. | | showcollectionresource | “y”, “n” | Y | Indicates whether it needs to show resource list associated with business connect services. By default, it returns no resources | | billIndex | 0, -1, -2, -3 | Y | Bill index values to allow fetching previous bills. 0 is default, corresponds to latest bill. | | sourceid | 101, 102, 1001, 130, 119 |  | 101 -> CORE AB  102 -> CORE BC  1001 -> Enabler  130 -> KB  119 -> TQ CORE |   Examples:  account/2343465/sub/4169999999/subscriber-summary |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given parameters |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "subscriberSummaryResponse" : {  "serviceSummary": {  "serviceInfo": <serviceInfo>,  "serviceInfo": {  billServiceInstanceId: integer,  serviceTypeCd: string,  primaryServiceResourceId: string,  serviceInstanceSeqNum: integer,  serviceUserName: string,  serviceCollectionTypeCd: string,  serviceCollectionGroupId: integer,  serviceCollectionResourceList: [<serviceCollectionResource>]  },  {  “preTaxChargeAmt”: Decimal,  “monthlyChargeAmt”: Decimal,  “monthlyChargeItemList”: [<chargeItem>],  “partialChargeAmt”: Decimal,  “partialChargeItemList”: [<chargeItem>],  “addOnAmt”: Decimal,  “addOnList”: [<chargeItem>],  “taxAmt”: Decimal,  “totalChargeAmt”: Decimal,  “savingAmt”: Decimal  },  “usageChargeTotalAmt”: decimal,  “usageChargeList”: [<usageCharge>],  “taxItemList”: [<taxDisplay>]  },  "status” : <status>  }  Where “serviceInfo” is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | billServiceInstanceId | integer | Bill Service Instance ID |  | “1” | | serviceTypeCd | String | Type of service with possible values, “MOBILE” (wireless), “INTERNET” (High speed internet), “TV” (Telus TV), “LANDLINE” (home phone), and “BUSINESS\_SERVICE” (business service), “BUSINESS\_CONNECT” (Business Connect Seat) |  | “MOBILE”, “INTERNET”, “TV”, “LANDLINE”, “BUSINESS\_CONNECT”, “BUSINESS\_SERVICE” | | primaryServiceResourceId | String | Cell number for wireless service | Y | “4169999999” | | serviceInstanceSeqNum | Integer | Service Instance Sequence number | Y | “0” | | serviceUserName | String | Service user name | Y | “DARCY MCGUIRE” | | serviceCollectionTypeCd | String | Business Connect service type | Y | “STARTER”, “OFFICE”,“MOBILE”, “PROFESSIONAL” | | serviceCollectionGroupId | Integer | Business Connect group location Id | Y | “54594” | | serviceCollectionResourceList | List | List of service collection resources | Y | [<serviceCollectionResource>] |   Where <serviceCollectionResource> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | resourceId | Integer | Resource Id |  | WLS | “4542123” |  | | resourceTypeCd | String | Resource type code |  | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | displayTxt | String | Invoice display text |  | WLS | “Toll Free number” |  | | resourceValueTxt | String | The identifier for the resource, i.e. phone number |  | WLS | “18665452365” |  |   Where “serviceCharge” is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | preTaxChargeAmt | decimal | Pre-tax total service Amount |  | “120.00” | | partialChargeAmt | decimal | Partial charges | Y | “120.00” | | monthlyChargeAmt | decimal | Monthly/Regular Charges |  | “120.00” | | addOnAmt | decimal | Add-on Amount | Y | “120.00” | | taxAmt | decimal | Tax on charges |  | “120.00” | | totalChargeAmt | decimal | Total Charges tax included |  | “120.00” | | savingAmt | decimal | Saving Amount | Y | “120.00” | | usageChargeTotalAmt | decimal | Total usage summary charge | Y | “120.00” | | monthlyChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y | <chargeItem> | | partialChargeItemList | List | A list of detail charge items, will be shown based on input showLineItem | Y | <chargeItem> | | addOnList | List | A list of detail charge items, will be shown based on input showLineItem | Y | <chargeItem> |   Where <chargeItem> is:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Attr**  **Type** | **Possible/typical values** | **Decoration** | | chargeItemDescriptionTxt | decimal | Description |  |  | “Paper bill Fee” |  | | chargeItemAmt | decimal | Balance forward from last bill |  |  | “120.00” |  | | chargeItemAmtTxt | String | Text representation of the charge item amount. i.e. "FREE" , "0" , “B“ (which means suppress amount) , or the exact value as chargeItemAmt |  | FFH | “120.00”, “0”, “Free”, “B” |  | | servicePackageGroupCd | String | Business Connect attribute showing the resource type associated with the charge | Y | WLS | “MOBILE”, “VOIP”, “HSIA”, “TOLL\_FREE”, “LOCAL” |  | | bundleChargeItemSubList | List | A list of the sub items <chargeItem> as bundle offer | Y | FFH | <chargeItem> | N |   Where <usageCharge> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | **Decoration** | | usageSummaryTypeCd | String | Type of usage summary with possible values, “AIRTIME”, “DATA”, “OTHER”, “ON-DEMAND”, “PAY-PER-VIEW”, mapped to the usage summary type code in billing ODS | “AIRTIME”, “DATA”, “OTHER”, “TV”, “INTERNET”, “LANDLINE”, “VOIP” |  | | usageSummaryTxt | String | Usage summary description text | “Airtime”, “Data”, “Other”, “TV”, “INTERNET”, “LANDLINE” | X | | usageSummaryChargeAmt | decimal | Usage summary charges | “120.00” |  |   Where <taxDisplay> is   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | displayTxt | String | Tax Display text | “GST / HST”, “GST”, “PST” | | taxAmt | decimal | Tax amount | “12.34” | | taxTypeCd | String | Tax type code | “GHST”, “HST”, “PST” | | taxAuthCd | String | Tax authorization code | “GOV” |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "subscriberSummaryResponse": {  "serviceSummary": {  "serviceInfo": {  "billServiceInstanceId": 13423423,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992929",  "serviceInstanceSeqNum": 10,  "serviceUserName": "Kevin Lo"  },  "serviceCharges": {  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "partialChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "monthlyChargeAmt": 100,  "monthlyChargeItemList": [  {  "chargeItemDescriptionTxt": "Voice 20",  "chargeItemAmt": 20  },  {  "chargeItemDescriptionTxt": "911 Charge",  "chargeItemAmt": 2  }  ],  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  },  "usageChargeTotalAmt": 45.5,  "usageChargeList": [  {  "usageSummaryTypeCd": "DATA",  "usageSummaryTxt": "Data",  "usageSummaryChargeAmt": 25.5  },  {  "usageSummaryTypeCd": "VOICE",  "usageSummaryTxt": "Airtime",  "usageSummaryChargeAmt": 20  },  {  "usageSummaryTypeCd": "OTHER",  "usageSummaryTxt": "Others",  "usageSummaryChargeAmt": 20  }  ]  },  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 2000 ms, 4000 ms peak |

## bill-archives

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-archives  sourceid/{sourceid}/account/{ban}/bill-archives | |
| **Method** | GET |
| **Description** | Return the list of invoice archives available |
| **Input** | **ban** – account number (currently only wireless postpaid, compass (wireline consumer and single line business) supported  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Type** | **Comments** | | system | "wls", "ffh" | Y |  | Billing system – default is wireless  **Note: This is not required in v3.0 as the sourceid determines the billing system source** | | start | 1 | Y |  | By default, it returns last 18 months starting from latest bill. Once this is set, it returns the bill archive list specified by the start and end index by months in reference to the latest bill. NOTE: The maximum of end index is 18 | | end | 3 | Y |  | By default, it returns last 18 months including latest bill. Once this is set, it returns the bill archive list specified by the start and end index by months in reference to the latest bill. NOTE: The maximum of end index is 18 | | pending | “y”,”n” | Y | WLS | By default, it is “n”, it returns latest released bill. If it is “y”, the result will show pending bill. | | sourceid | 101, 102, 1001, 130, 119 |  |  | 101 -> CORE AB  102 -> CORE BC  1001 -> Enabler  130 -> KB  119 -> TQ CORE |   Examples:  account/2343465/bill-archives  account/2343465/bill-archives?start=1&end=3 |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given BAN |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billArchivesResponse" : {  "billArchiveList": [ <billArchiveInfo> ],  "status” : <status>  }  Where <billArchiveInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | locatorId | String | Bill Locator ID |  | E2V\_WLSPROD|123456789|2013-12-17|qwerty | | bill**Dt** | Date | Bill date |  | 2013-12-19-04:00 | | generationStatusCd | String | Bill archive generation status code | Y | “READY”, “LOADING”, “PENDING” |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "billArchivesResponse": {  "billArchiveList": [  {  "locatorId": "E2V\_WLSPROD|123456789|2013-04-25|qwerty",  "billDt": "2013-12-15",  "generationStatusCd": "READY"  },  {  "locatorId": "E2V\_WLSPROD|123456789|2013-10-26|qwerty",  "billDt": "2013-11-15",  "generationStatusCd": "LOADING"  },  {  "locatorId": "E2V\_WLSPROD|123456789|2013-11-27|qwerty",  "billDt": "2013-10-15",  "generationStatusCd": "PENDING"  }  ],  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 2000 ms, 4000 ms peak |

## bill-archive-url

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-archive-url  sourceid/{sourceid}/account/{ban}/bill-archive-url | |
| **Method** | GET |
| **Description** | Return the actual download URL for the given bill locator ID |
| **Input** | **ban** – account number (currently only wireless postpaid, compass (wireline consumer and single line business) supported  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Comments** | | system | "wls", "ffh" | Y | Billing system – default is wireless  **Note: This is not required in v3.0 as the sourceid determines the billing system source** | | locatorid | E2V\_WLSPROD|123456789|2013-12-17|qwerty |  | This is a unique ID retrieved for each bill archive from bill-archive-list operation | | brandname | “TELUS”, “KOODO” | Y | Brand name | | sourceid | 101, 102, 1001, 130, 119 |  | 101 -> CORE AB  102 -> CORE BC  1001 -> Enabler  130 -> KB  119 -> TQ CORE |   Examples:  account/2343465/bill-archive-url?locatorid=E2V\_WLSPROD|123456789|2013-12-17|qwerty |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given locator ID |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billArchiveUrlResponse”: {  < billArchiveUrlInfo>,  "status” : <status>  }  Where <billArchiveUrlInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Optional** | **Possible/typical values** | | url | String | Bill download URL |  | http://um-billpresent-west-pr.tsl.telus.com/CMO…. | | generationStatusCd | String | Bill archive generation status code | Y | “READY”, “LOADING”, “PENDING” |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "billArchiveUrlResponse": {  "url": "http://um-billpresent-west-pr.tsl.telus.com/CMO/BillingInquiryMgmt/BillDocumentRetrievalServlet/download/2794383\_2013-10-08.pdf?document=l/5N5ydxW%2BlQmpsMI/HrG8l96EwD6i54BBzHVX1cvnpA64nkMmQSAtuIk3KrqFmvHySv6Aw4YPi%2BW6SKiPzpHbY/hxtu0SO2L0PInaB1Kuo=",  “generationStatusCd”: “READY”,  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 2000 ms, 4000 ms peak |

## video-bill

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/video-bill | |
| **Method** | GET |
| **Description** | Return the video bill url generated by the third party vendor |
| **Input** | **ban** – account number (currently only wireless postpaid are supported)  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Comments** | | system | "wls", "ffh" | Y | Billing system – default is wireless | | uuid | c61dbbc2-77a1-4095-8668-60b01e7cccfc | Y | UUID from identity | | useragent | “Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.125 Safari/537.36” |  | This is to identify the type of user’s browser | | billdt | “2014-06-30” | Y | To retrieve video-bill with specific bill date | | autopayment | “y”, “n” | Y | Indicate whether the account is on autopayment or not | | customername | “Kevin” | Y | First name of the video bill user | | debug | “y”, “n” | Y | Print out the SundaySky payload as trace in the response | | videourl | “y”, “n” | Y | If false, then service will not generate sundaySky URL rather respond with a dummy URL. This is built for debug purpose and to avoid unnecessary request to Sundaysky |   Examples:  account/2343465/video-bill?system=wls&uuid=c61dbbc2-77a1-4095-8668-60b01e7cccfc |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given BAN |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "videoBillResponse" : {  "url": string,  “trace”: string,  "status” : <status>  }  Where <billArchiveInfo> is:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | url | String | The video bill URL generated by third party vendor |  | http://ss.com/flv?token=347289472389479 |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "videoBillResponse": {  "url": “http://ss.com/flv?token=347289472389479”,  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 3000 ms, 5000 ms peak |

## bill-delivery-preference

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-delivery-preference | |
| **Method** | GET |
| **Description** | Return the billing method, and bill notification preferences for eBill (email/SMS) |
| **Input** | **ban** – account number  **uuid** – Identity user id  Parameters   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Type** | **Comments** | | system | "wls", "ffh" | Y |  | Billing system – default is wireless | | latestchange | “y”, “n” | Y | WLS | By default, it will retrieval all records including history | | uuid | “c61dbbc2-77a1-4095-8668-60b01e7cccfc” | Y | WLS | To retrieve a specific bill delivery preference for a particular user |   Examples:  account/2343465/bill-delivery-preference?system=wls&latest=y |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given BAN |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "billDeliveryPreferenceResponse" : {  “notificationPreferences”: [<notificationAddress>]  "status” : <status>  }  Where notificationAddress is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Attr Type** | **Opt** | **Possible/typical values** | | deliveryTypeCd | String | Bill delivery type |  |  | “EBILL”, “PAPER”, “EPOST” | | addressTypeCd | String | Address type |  | Y | “SMS”, “EMAIL” | | addressTxt | String | Email/SMS address |  | Y | “[john.woe@telus.com](mailto:john.woe@telus.com)”, “4169991010” | | userId | String | UUID/CustomerId |  |  | “c61dbbc2-77a1-4095-8668-60b01e7cccfc”, “4545644” | | effStartDt | Date | Effective start date |  | Y | “2013-04-08-04:00” | | effEndDt | Date | Effective end date | WLS | Y | “2014-04-08-04:00” | | validationStatusCd | String | Indication whether the address is validated or not | WLS | Y | “V”/“P” = validated/pending |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  " billDeliveryPreferenceResponse": {  " notificationPreferences": [  {  deliveryTypeCd: “EBILL”,  addressTypeCd: “SMS”,  addressTxt: “4169394949”  userId: “c61dbbc2-77a1-4095-8668-60b01e7cccfc”,  effStartDt: “2014-04-08-04:00”,  validationStatusCd: “V”  },  {  deliveryTypeCd: “EBILL”,  addressTypeCd: “EMAIL”,  addressTxt: “joe.bow@telus.com”  userId: “c61dbbc2-77a1-4095-8668-60b01e7cccfc”,  effStartDt: “2014-04-08-04:00”,  validationStatusCd: “V”  },  ]  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 3000 ms, 5000 ms peak |

## update-bill-delivery-preference

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/bill-delivery-preference | |
| **Method** | POST |
| **Description** | Return the billing method, and bill notification preferences for eBill (email/SMS) |
| **Input** | **ban** – account number  The parameters will be passed in the header of the HTTP request within a JSON structure, e.g.  {  "notificationPreference": [<NotificationPreference>]  }  "system": string  “ban”: string  <NotificationPreference> can be described as:  {  actionType: string,  notificationAddresses: [<notificationAddress>]  }  Parameter:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Attr Type** | **Opt** | **Possible/typical values** | | actionType | String | Action on the notification setting |  |  | “register” - Bill delivery type must be EBILL or EPOST  “deregister” – with Bill delivery type of “PAPER” => turn off ebill and turn on paper bill | | system | string | Billing system – default is wireless |  | Y | “ffh”, “wls” | | ban | String | Billing account number |  |  | “3434232” |   Where notificationAddress is:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Attr Type** | **Opt** | **Possible/typical values** | | deliveryTypeCd | String | Bill delivery type |  |  | “EBILL”, “PAPER”, “EPOST” | | addressTypeCd | String | Address type |  | Y | “SMS”, “EMAIL” | | addressTxt | String | Email/SMS address |  | Y | “[john.woe@telus.com](mailto:john.woe@telus.com)”, “4169991010” | | userId | String | UUID/CustomerId |  |  | “c61dbbc2-77a1-4095-8668-60b01e7cccfc”, “4545644”  Note: not mandatory with calls via SDF | | effStartDt | Date | Effective start date |  | Y | “2013-04-08-04:00” | | effEndDt | Date | Effective end date | WLS | Y | “2014-04-08-04:00” | | validationStatusCd | String | Indication whether the address is validated or not | WLS | Y | “V”/“P” = validated/pending | |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given BAN |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "updateBillDeliveryPreferenceResponse" : {  "status” : <status>  } |
| **SLA** | Expected average response time: 3000 ms, 5000 ms peak |

## group-summary

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/group-summary | |
| **Method** | GET |
| **Description** | Return the latest list of group summary with only subscriber level summarized information for the given BAN/CAN with optional search criteria |
| **Input** | **ban** – account number (currently only wireless postpaid) supported  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Opt** | **Comments** | | totalservicecount | “300” | Y | passed as total count of services for latest invoice if known.  used as optional optimization parameter to avoid aditional query to backend. | | start | “1” | Y | start index of service in case limited report items are to be fetched  works in conjunction with end parameter. 'start' less than 'end' should be true else 'start' parameter will reset to 1 | | end | “200” | Y | end index of service in case limited report items are to be fetched  works in conjunction with start parameter. 'start' less than end should be true else end parameter will be reset to total service count.  default value is total service count. |   Examples:  account/2343465/group-summary  account/2343465/group-summary?start=1&end=400 |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given parameters |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "groupSummaryResponse" : {  "groupSummaryList": [  {  billServiceInstanceId: integer,  serviceTypeCd: string,  primaryServiceResourceId: string,  serviceUserName: string,  preTaxChargeAmt: decimal,  partialChargeAmt: decimal,  monthlyChargeAmt: decimal,  addOnAmt: decimal,  usageChargeTotalAmt: decimal,  taxAmt: decimal,  totalChargeAmt: decimal,  savingAmt: decimal  }  ],  "status” : <status>  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | billServiceInstanceId | integer | Bill Service Instance ID |  | “1” | | serviceTypeCd | String | Type of service with possible values, “MOBILE” (wireless), “BUSINESS\_CONNECT” (Business Connect Seat) |  | “MOBILE”, “BUSINESS\_CONNECT” | | primaryServiceResourceId | String | Cell number for wireless service | Y | “4169999999” | | serviceUserName | String | Service user name | Y | “DARCY MCGUIRE” |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Type** | **Possible/typical values** | **Decoration** | | preTaxChargeAmt | decimal | Pre-tax total service Amount |  |  | “120.00” |  | | partialChargeAmt | decimal | Partial charges | Y |  | “120.00” |  | | monthlyChargeAmt | decimal | Monthly/Regular Charges |  |  | “120.00” |  | | addOnAmt | decimal | Add-on Amount | Y | WLS | “120.00” |  | | taxAmt | decimal | Tax on charges |  | WLS | “120.00” |  | | totalChargeAmt | decimal | Total Charges tax included |  | WLS | “120.00” |  | | savingAmt | decimal | Saving Amount | Y | WLS | “120.00” |  | | usageChargeTotalAmt | decimal | Total usage summary charge | Y |  | “120.00” |  |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | | {  "groupSummaryResponse": {  "groupSummaryList": [  {  "billServiceInstanceId": 13423423,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992929",  "serviceUserName": "Kevin Lo"  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  "usageChargeTotalAmt": 45.5  },  {  "billServiceInstanceId": 13423410,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992119",  "serviceUserName": "Zhong Ke"  "preTaxChargeAmt": 30,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 150.5,  "savingAmt": 10  "usageChargeTotalAmt": 65.5,  },  {  "billServiceInstanceId": 13423499,  "serviceTypeCd": "MOBILE",  "primaryServiceResourceId": "4168992519",  "serviceUserName": "Jim Cai"  "preTaxChargeAmt": 100,  "partialChargeAmt": 50.5,  "monthlyChargeAmt": 100,  "addOnAmt": 20,  "taxAmt": 150.5,  "totalChargeAmt": 300,  "savingAmt": 10  "usageChargeTotalAmt": 0  }  ],  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorCd": null,  "systemErrorTimeStamp": null,  "systemErrorTxt": null  }  }  } |  | |
| **SLA** | Expected average response time: 6000 ms |

## restricted-tax-credit-report

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/restricted-tax-credit-report | |
| **Method** | GET |
| **Description** | Return the list of telecom restricted tax credit report for the given BAN. |
| **Input** | **ban** – account number (only wireless postpaid) supported  Examples:  account/2343465/restricted-tax-credit-report |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 |  | No information found for the given parameters |  |  |  | | 500 |  | general error | underlying error code | underlying error message | Any caught exception not captured elsewhere | |
| **Output** | "getRestrictedTaxCreditReportResponse" : {  "restrictedTaxCreditReportList": [  {  taxEligibleYear: string,  "taxSupplementarySummaryList": [<TaxSupplementarySummary>  ],  "status” : <status>  }  Where <TaxSupplementarySummary> as  {  billDt: date,  totalAmt: decimal,  "taxSupplementaryDetailList":[<TaxSupplementaryDetail>]    }  Where <TaxSupplementaryDetail> as  {  billDt: date,  provinceCd: string,  taxSupplementaryDetailDescriptionTxt: string,  taxPortionAmt: decimal  }   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Opt** | **Possible/typical values** | | taxEligibleYear | integer | Tax year |  | 2015 | | billDt | String | Bill date |  | 2015-04-14 | | totalAmt | decimal | Tax total amount | Y | 123.12 | | provinceCd | String | Province code | Y | “ON” | | taxSupplementaryDetailDescriptionTxt | String | Tax description |  |  | | taxPortionAmt | decimal | Tax portion amount |  | 23.12 |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |   Example:   |  |  | | --- | --- | | **Undecorated** | **Decorated** | |  |  | |
| **SLA** | Expected average response time: 6000 ms |

# Appendix

## Error Codes

|  |  |
| --- | --- |
| IB | Invalid BAN |
| IS | Invalid Subscriber Number |
| ISI | Invalid Subscriber ID |
| IBC | Invalid Bill Cycle |
| ID | Invalid Date |
|  |  |

## Decommissioned Services

UsageManagement 1.0, 2.0

UsageNotifications