Open API Project

USCC AMP Subscriber Profile REST Developer Guide

Document Version 1.3

Document Revision History

Rev #	Date	Description
1.0	June 28 2013	Created based on USCC-NaaS-Sub-Profile_API_v1.0.doc.
1.1	Sept 12 2013	Added Sandbox section as an Appendix.
1.2	Nov 1 2013	Replaced URLs, code examples, param lists to match production platform.
1.3	Mar 31 2015	Updated for Prepaid Flag CR. Sections 4.3.1, 4.3.2, 4.4.2 and A2.1

Legal Information

Table of Contents

1 Subscriber Profile REST Overview

The U.S. Cellular Open API User Subscriber Profile web service allows an application to query subscriber profile information of one or more U.S. Cellular subscribers identified by their CTN. The service is accessible via the RESTful API, which is described in this document.

2 Authentication

Connection to the US Cellular AMP Subscriber Profile web service by a TPA is made using one-way TLS, where a server side certificate is required plus HTTP Basic Authentication.

3 Methods

The following method is available:

- retrieve subscriber profile of one or more subscribers – section 4.1

HTTP GET command is used.

3.1 URIs

The URI of the resource is as follows:

https://developerportal.uscellular.com/cxf/subscriberProfile/{apiVersion}/?address={address}

The following standard URL variables apply:

Name	Description
apiVersion	The Version of the API that the client is accessing: in this case, v1.
address	Address (CTN) of U.S. Cellular subscriber, in the format comprising: 1) 'tel:' protocol identifier, 2) country code of one preceded by '+', 3) ten digit CTN preceded by 1, for example, tel:+15087300001

	Finally, the address must be must be URL-escaped where %3A represents ':' and %2B represents '+'. Example: %3A%2B15087300001

4 Get Subscriber Profile

4.1 Retrieve Subscriber Profile Request

This operation is used by the TPA to request the subscriber profile of one or more U.S. Cellular subscribers.

4.1.1 Request Example – Single Subscriber

```
GET https://developerportal.uscellular.com/cxf/subscriberProfile/v1/?address=tel3A
%2B15087300001

HTTP/1.1
Accept: application/json

Host: developerportal.uscellular.com

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl
```

4.1.2 Request Example – Multiple Subscribers

```
GET https://developerportal.uscellular.com/cxf/subscriberProfile/v1/?address=tel3A
%2B15087300001&address=tel3A%2B15087300002 HTTP/1.1

Accept: application/json

Host: developerportal.uscellular.com

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl
```

4.2 Request Parameters

Parameter	Data Type	Description	Optional
address	String	Address (CTN) of U.S. Cellular subscriber, in the format comprising:	No; one or more

		1) 'tel:' protocol identifier, 2) country code of one preceded by '+', 3) ten digit CTN preceded by 1, for example, tel:+15087300001 Finally, the address must be must be URL-escaped where %3A represents ':' and %2B represents '+'. Example: %3A%2B15087300001	

4.3 Retrieve Subscriber Profile Responses

The response content type for the Subscriber Profile API is application/json.

As a general practice, null elements are not returned in the responses.

The following types of responses are listed in this section:

- Successful response with subscriber profile for one CTN returned - section below
- Successful response with some of the subscriber profiles for the CTNs returned - section 4.3.2
- Unsuccessful response with no external call made to retrieve subscriber profile - 4.3.3

Error information structures are summarised diagrammatically in section 4.5.

4.3.1 Response Example – One CTN - Success

```
HTTP/1.1 200 OK
Date: Thu, 31 Oct 2013 09:03:44 GMT
Server: Jetty(7.5.4.v20111024)
Content-Type: application/json
Content-Length: 684
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
{
```



```
 "subscriberProfileList" : [ {
  "address" : "tel:+15087300001",
  "profileRetrievalStatus" : "RETRIEVED",
  "inventoryData" : {
    "prepaidFlag”: true,
    “subscriberHandsetSalesChannelCapableList" : {
     "shscc" : [ "YES" ]
    },
    "subscriberHomeSwitchTimeZoneList" : {
     "shstz" : [ "6" ]
    },
    "subscriberStatusList" : {
     "ss" : [ "ACTIVE" ]
    },
    "subscriberAccountTypeList" : {
     "sat" : [ "I" ]
    },
    "subscriberAccountSubTypeList" : {
     "sast" : [ "R" ]
    },
    "subscriberPrimaryMDNindList" : {
     "spmi" : [ "N" ]
    },
    "subscriberAccountList" : {
     "sa" : [ "123456789" ]
    }
  }
 } ]
}
```


4.3.2 Response Example – Multiple CTNs - Partial Success

The example below returns some of the subscriber profiles successfully, and others not, as indicated by the profileRetrievalStatus parameter value.

```
HTTP/1.1 200 OK
Date: Thu, 31 Oct 2013 09:03:44 GMT
Server: Jetty(7.5.4.v20111024)
Content-Type: application/json
Content-Length: 684
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
{
  "subscriberProfileList" : [
   "address" : "tel:+16309700001",
    "profileRetrievalStatus" : "NOT_RETRIEVED",
    "errorInformation" : {
      "messageId" : "SVC0005",
      "text" : "An external service error occurred. No profile information is available for %1",
     "variables" : [ "tel:+16309700001" ]
  }
}, {
    "address" : "tel:+15087300001",
    "profileRetrievalStatus" : "RETRIEVED",
    "inventoryData" : {
      "subscriberHandsetSalesChannelCapableList" : {
        "shscc" : [ "YES" ]
      },
      "subscriberHomeSwitchTimeZoneList" : {
        "shstz" : [ "6" ]
      },
      "prepaidFlag”: true,
```



```
    "subscriberStatusList" : {
      "ss" : [ "ACTIVE" ]
    },
    "subscriberAccountTypeList" : {
      "sat" : [ "I" ]
    },
    "subscriberAccountSubTypeList" : {
      "sast" : [ "R" ]
    },
    "subscriberPrimaryMDNindList" : {
      "spmi" : [ "N" ]
    },
    "subscriberAccountList" : {
      "sa" : [ "123456789" ]
    }
  }
} , {
  "address" : "tel:+15087300002",
  "profileRetrievalStatus" : "RETRIEVED",
  "inventoryData" : {
    "subscriberHandsetSalesChannelCapableList" : {
      "shscc" : [ "YES" ]
    },
    "subscriberHomeSwitchTimeZoneList" : {
      "shstz" : [ "6" ]
    },
    "prepaidFlag”: false,
    "subscriberStatusList" : {
      "ss" : [ "ACTIVE" ]
    },
    "subscriberAccountTypeList" : {
      "sat" : [ "I" ]
```



```
    },
    "subscriberAccountSubTypeList" : {
      "sast" : [ "E" ]
    },
    "subscriberPrimaryMDNindList" : {
      "spmi" : [ "Y" ]
    },
    "subscriberAccountList" : {
      "sa" : [ "999996789" ]
    }
   }
  }, {
   "address" : "tel:+1234567890",
   "profileRetrievalStatus" : "ERROR",
   "errorInformation" : {
    "messageId" : "SVC0001",
    "text" : "A service error occurred."
   }
}]
}
```


4.3.3 Response Example – Unsuccessful

The exception in this unsuccessful example was thrown before Aepona made the external call to retrieve the required subscriber profile. Therefore no subscriber profile content is returned. Policy violations can result in exceptions of this type. See section 5.2 for the list of policy and service error codes.

```
HTTP 400 Bad Request
Content-Type: application/json
Content-Length: 1234
Date: Thu, 04 Jun 2009 02:51:59 GMT
```



```
{"requestError": {
   "serviceException": {
      "messageId": "SVC0004",
      "text": "No valid addresses provided in message part %1",
      "variables": "%1 - request URI"
   }
}}
```

4.4 Response Parameters

Successful responses are returned using the following hierarchy of data structures:

subscriberProfileList – an array, containing one or more

---*subscriberProfile* element – containing subscriber profile information per CTN, containing

----inventoryData element - with optional

----sub-elements - as supplied by the ESB, each of which can contain one or more optional

----sub-element

----ProfileRetrievalStatus element – indicates status: Retrieved, Not Retrieved, with error information, or Error (service or policy exception)

The details of each structure and sub-elements are described in the following sections.

4.4.1 subscriberProfileList /subscriberProfile Elements

Parameter	Data Type	Description	Usage
subscriberProfileList	Object [..1]	All of the following parameters are nested within 'subscriberProfleList'	If the subscriber profile information is successfully retrieved for a least one address (CTN), then this element must be included. If the subscriber profile information isn't successfully retrieved

Parameter	Data Type	Description	Usage
			for any of the addresses, then a non-200 HTTP response code is returned, with a Service or Policy exception (see section 5.2) as applicable.
subscriberProfile	Object [1..N]	This element is an array where each element in the array either contains a) if successfully retrieved, the Subscriber Profile information for a given address (CTN) OR b) if not successfully retrieved, the error information. This object contains the following elements: 1) 'address' 2) 'profileRetrievalStatus' 3) either a) the 'inventoryData', if the profile information is successfully retrieved for the specified 'address' OR b) 'errorInformation' if there is an error in profile retrieving the information for the specified 'address'	If 'subscriberProfileList' object is included, then this object is mandatory.
address	String	The CTN of the U.S. Cellular subscriber. See description	If 'subscriberProfile' object is included, then

Parameter	Data Type	Description	Usage
		and example in section 4.1.	this object is mandatory.
inventoryData	Object	For a given address (CTN), it contains the Subscriber profile information. This object's elements are defined in section 4.4.2.	If the profile information is successfully retrieved for the specified 'address', then this object is mandatory
errorInformation	Object	For a given address (CTN), it contains the error information. This object's elements are defined in section 4.4.3.	If the profile information is not successfully retrieved for the specified 'address', then this object is mandatory
profileRetrievalStatus	String	If the profile information is successfully retrieved for the specified 'address', then this element's value is RETRIEVED. If the profile information is not successfully retrieved, then this element's value is NOT_RETRIEVED (when an exception is returned from the external subscriber profile provider). If an internal error or external subscriber profile provider timeout occurs, then the value ERROR is returned.	If 'subscriberProfile' object is included, then this object is mandatory.

4.4.2 inventoryData Structure Elements

The table below contains the full list of elements in this structure that may be returned in the response. Of these, the ones currently sent by USCC are noted in black characters; those not sent by USCC are in grey.

Parameter	Data Type	Description	Usage
subscriberHandsetSalesChannelCapableList	Object	Defines whether or not a wireless device is capable of supporting a service on it. This element contains the 'subscriberHandsetSalesChannelCapable' element.	An optional element of 'inventoryData' object.
subscriberHandsetSalesChannelCapable	String [1.. unbounded]	This element is an array where each element in the array has a value of one of the following: a) YES b) NO c) UNKNOWN	If 'subscriberHandsetSalesChannelCapableList' object is included, then this object is mandatory, one or more.
subscriberHomeSwitchTimeZoneList	Object	Provides information on the subscriber's home switch time zone. This element contains the 'subscriberHomeSwitchTimeZone' element.	An optional element of 'inventoryData' object.
subscriberHomeSwitchTimeZone	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberHomeSwitchTimeZoneList' object is included, then this object is mandatory, one ore more.

prepaidFlag	Boolean	Provides information on the subscriber's account type: a) true – Prepaid b) false – not Prepaid	An optional element of 'inventoryData' object. Information not available will return an SVC0005 error.
subscriberStatusList	Object	Provides information on the subscriber's status. This element contains the 'subscriberStatus' element.	An optional element of 'inventoryData' object.
subscriberStatus	String [1.. unbounded]	This element is an array where each element in the array has a value of one of the following: a) ACTIVE b) CANCELLED c) RESERVED d) SUSPENDED e) UNKNOWN	If 'subscriberStatusList' object is included, then this object is mandatory, one or more.
subscriberAccountTypeList	Object	Provides information on the subscriber's account type. This element contains the 'subscriberAccountType' element.	An optional element of 'inventoryData' object.
subscriberAccountType	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberAccountTypeList' object is included, then this object is mandatory, one or more.

subscriberAccountSubTypeList	Object	Provides information on the subscriber's account sub-type. This element contains the 'subscriberAccountSubType' element.	An optional element of 'inventoryData' object.
subscriberAccountSubType	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberAccountSubTypeList' object is included, then this object is mandatory, one or more.
subscriberPrimaryMDNindList	Object	Provides information on the subscriber's MDN indicator. This element contains the 'subscriberPrimaryMDNind' element.	An optional element of 'inventoryData' object.
subscriberPrimaryMDNind	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberPrimaryMDNindList' object is included, then this object is mandatory, one or more.
subscriberAccountList	Object	Provides information on the subscriber's account. This element contains the 'subscriberAccount' element.	An optional element of 'inventoryData' object.
subscriberAccount	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberAccountList' object is included, then this object is mandatory, one or more.

subscriberFullNamList	Object	Provides information on the subscriber's name. This element contains the 'subscriberFullNam' element.	An optional element of 'inventoryData' object.
subscriberFullNam	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberFullNamList' object is included, then this object is mandatory, one or more.
subscriberMDNList	Object	Provides information on the subscriber's MDN. This element contains the 'subscriberMDN' element.	An optional element of 'inventoryData' object.
subscriberMDN	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberMDNList' object is included, then this object is mandatory, one or more.
subscriberMEIDList	Object	Provides information on the subscriber's MEID. This element contains the 'subscriberMEID' element.	An optional element of 'inventoryData' object.
subscriberMEID	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberMEIDList' object is included, then this object is mandatory, one or more.
subscriberMSIDList	Object	Provides information on the subscriber's MSID. This element contains the 'subscriberMEID' element.	An optional element of 'inventoryData' object.

subscriberMSID	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberMSIDList' object is included, then this object is mandatory, one or more.
subscriberESNList	Object	Provides information on the subscriber's ESN. This element contains the 'subscriberMEID' element	An optional element of 'inventoryData' object.
subscriberESN	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberESNList' object is included, then this object is mandatory, one or more.
subscriberBillingCycleList	Object	Provides information on the subscriber's billing cycle. This element contains the 'subscriberBillingCycle' element.	An optional element of 'inventoryData' object.
subscriberBillingCycle	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberBillingCycleList' object is included, then this object is mandatory, one or more.
subscriberAccountAgeList	Object	Provides information on the subscriber's account age. This element contains the 'subscriberAccountAge' element.	An optional element of 'inventoryData' object.

subscriberAccountAge	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberAccountAgeList' object is included, then this object is mandatory, one or more.
subscriberHandsetAgeList	Object	Provides information on the subscriber's handset age. This element contains the 'subscriberHandsetAge' element.	An optional element of 'inventoryData' object.
subscriberHandsetAge	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberHandsetAgeList' object is included, then this object is mandatory, one or more.
subscriberBillingZipCodeList	Object	Provides information on the subscriber's billing zip code. This element contains the 'subscriberBillingZipCode' element.	An optional element of 'inventoryData' object.
subscriberBillingZipCode	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberBillingZipCodeList' object is included, then this object is mandatory, one or more.
subscriberRatePlanCodeList	Object	Provides information on the subscriber's rate plan code. This element contains the 'subscriberRatePlanCode' element.	An optional element of 'inventoryData' object.
subscriberRatePlanCod	String	This element is an array where	If

eCode	[1.. unbounded]	each element in the array is a string.	'subscriberRatePlanCodeList' object is included, then this object is mandatory, one or more.
subscriberServiceList	Object	Provides information on the subscriber's rate plan code. This element contains the 'subscriberService' element.	An optional element of 'inventoryData' object.
subscriberServiceCode	String [1.. unbounded]	This element is an array where each element in the array is a string.	If 'subscriberServiceList' object is included, then this object is mandatory, one or more.

4.4.3 errorInformation Structure Elements

Parameter	Data Type	Description	Usage
MessageId	String	Message identifier.	Mandatory element
Text	String	Message text, with replacement variables marked with %#, for example %1 for variable 1.	Mandatory element
Variable	String [0..unbounded]	Variables to substitute into Text string.	Optional element

4.5 Response Data Structure Diagrams

This section provides information on the exception scenarios and the structure of information returned in unsuccessful and successful responses. The labels are internal implementation java class names.

1. As illustrated in the example in section 4.3.2, a TPA JSON response can contain either a subscriberProfileList or a requestError. The diagram below shows the data structure of the SubscriberProfileReponse:

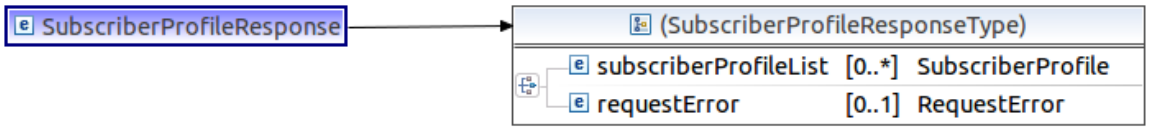

Figure 1: Response structure - with list or error returned

2. As illustrated in the example in section 4.3.3, an exception may be thrown before Aepona makes the call to the external service. The diagram below shows the exception structures in this case:

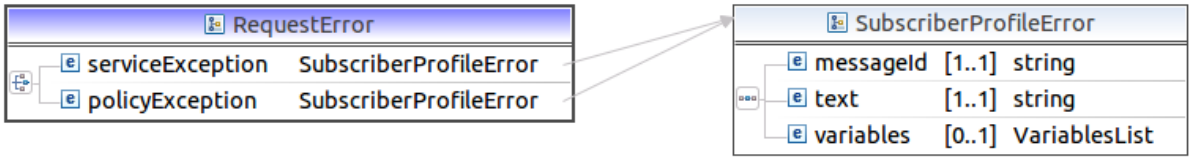

Figure 2: Response structure - with no external call made

3. The diagram below shows structures nested in SubscriberProfile data type (described in section 4.4.1). It shows that an entry in subscriberProfileList can contain error information from the external service, or inventoryData (described in section 4.4.2).

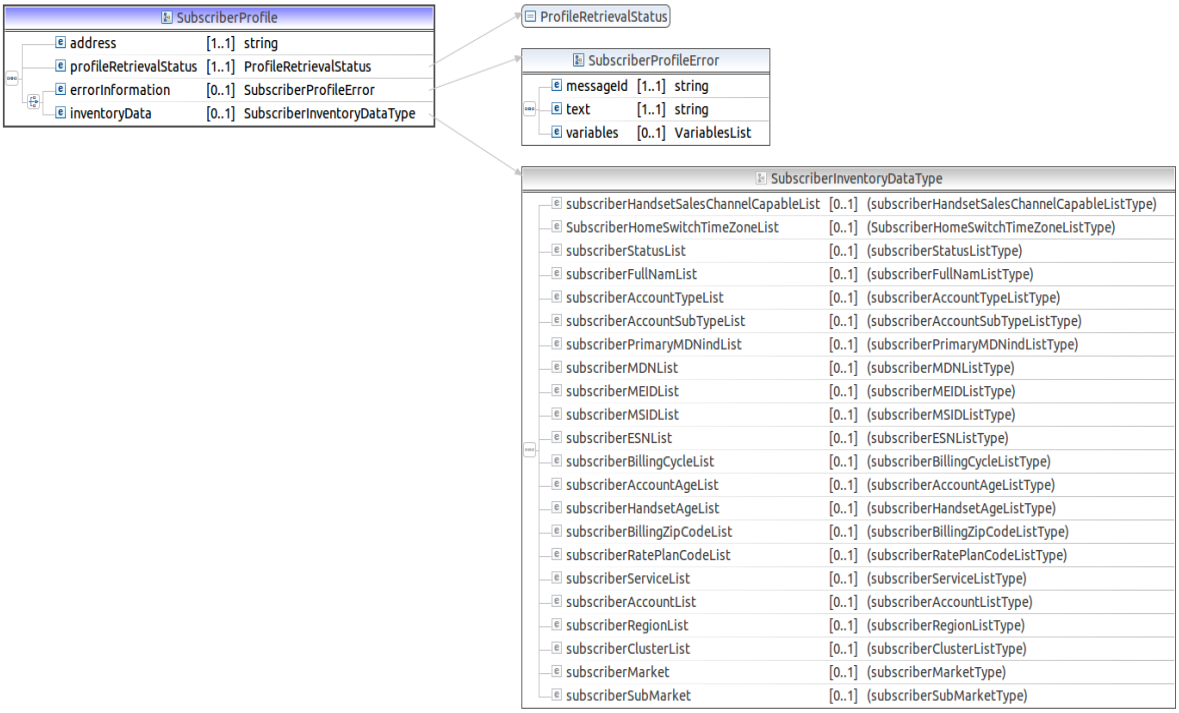

Figure 3: subscriberprofile data structure

5 Response Codes and Exceptions

5.1 HTTP Response Codes

Response Code	Explanation
200 OK	Success!
400 Bad Request	The request was invalid. An accompanying error message will explain why. For instance, the MSISDN specified in the request may not be a valid U.S. Cellular subscriber.
401 Unauthorized	The request requires user authentication and the authentication failed. Authentication credentials were missing or incorrect.
403 Forbidden	The request is understood, but it has been refused.
404 Not Found:	The URI requested is invalid or the resource requested does not exist. There is a mistake in the host or path of the service URI.
405 Method Not Allowed	The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. For example, one mistakenly used a HTTP POST to query for the subscriber profile instead of a HTTP GET, where only HTTP GET is supported.
500 Internal Server Error	The server encountered an unexpected condition, which prevented it from fulfilling the request.
503 Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. Please retry the request.

5.2 Exceptions

5.2.1 Service Exceptions

A service exception describes the reason why the service cannot accept the request.

The following service exceptions may be thrown:

Error	Explanation

SVC0001 – Service error occurred	An internal service-related error has occurred as a result of a client invocation on the service. This category can be used for implementation-specific errors. Contact the support team.
SVC0002 – Invalid input value	An input parameter value is not of the expected type. Check the parameter types and re-submit your request.
SVC0004 – Invalid Address Format	The requested terminal device address was supplied in the wrong format
SVC0005 – An external service error occurred.	The Service received an error from an external source. Contact the support team if this persists.
SVC0006 – Timeout error received.	The Service timed out waiting for a response from an external source. Contact the support team if this persists.

5.2.2 Policy Exceptions

A policy exception means that the request syntax is valid, however an operator policy has been broken.

POL0001 - Policy error occurred

The above exception may be thrown to indicate a fault relating to a policy associated with the service. This category can be used for implementation-specific errors such as:

Error Text	Explanation
POL-006: TPA exceeded its maximum allowed rate of transactions	The maximum rate of transactions is exceeded. Ensure that the rate of your requests is within the limits set up in your SLA, e.g. 10 TPS (Transactions Per Second).
POL-014: White List is enforced, and address is not in White List	A white list is enforced and the number is not in the white list. Check your SLA details.
POL-015: Black List is enforced, and address is in Black List	A black list is enforced and the number is in the black list. Check you SLA details.

POL-016: Max Requests is enforced, and max requests has been exceeded	The maximum number of requests for this service is exceeded. Contact the support team.
POL-017: Operation is not allowed	The method/operation is not supported in your current SLA. Check your SLA and use a method that is supported.
POL-040: Max Destination Addresses is enforced and maximum destination addresses has been exceeded	A maximum destination address limit is enforced and it has been exceeded. Check your SLA for the limit and re-submit your request.

A Sandbox Service

The sandbox service replicates real U.S. Cellular Open API User Subscriber Profile web service and returns various response objects, or 'canned responses', to pre-configured subscriber CTN values. It does not connect to any external interfaces. Developers can use this to test different scenarios of their application without connecting to the real subscriber profile service.

A.1 Sandbox Service Design

The Sandbox service flow is described below in outline:

1 Normal Subscriber Profile service is invoked up to the point where the external call is made to the USCC ESB Remote Endpoint.

2 At this point the Sandbox route will be invoked instead of the real USCC ESB Endpoint.

This flow ensures that all the validation logic defined in the normal service is executed inside the original service itself.

3 Once the sandbox route is being executed, it will generate the relevant response object based on the configured values against the CTN provided in the request.

The values are configured by the administrator as sandbox service properties.

A.1.1 Sandbox URI

The URI of the resource for sandbox service is as follows:

https://developerportal.uscellular.com/cxf/subProfileSandbox/{apiVersion}/sandbox?address={address}

Sandbox Request Example – Single Subscriber Profile

```
GET https://developerportal.uscellular.com/cxf/subProfileSandbox/v1/sandbox?address=tel%3A%2B15087300004 HTTP/1.1

Accept: application/json

Authorization: Basic QWVwb25hVfdghfdghdfghdfgWF6eHN3MjNl

Host: developerportal.uscellular.com
```


A.2 Sandbox Scenarios

Sandbox responses are driven by the CTN sent in the request. A response will be returned based on the validity of the request and the configured response against the CTN value.

- If the request is invalid then the normal service flow will reject the request and return the relevant response according to normal service specification.
- If the request is valid then the sandbox service will return a response based on the configured details against the CTN. The scenarios are listed in the section below.

A.2.1 Successful Scenarios

The pre-configured CTN and request URI syntax, and expected responses are summarised in the table below. The responses include an inventory data object with hard coded values and with the status relative to the configured CTN value.

An example success response is shown below. The structure is common across all responses.

> **!** The sandbox returns only a subset of the structural elements available for this service. As stated in section 4.4.2, the production endpoint USCC sends only those elements noted in black characters in the table, thus only those may be returned.

Sandbox Response Structure

```
{
 "subscriberProfileList" : [ {
  "address" : "tel:+15087300001",
  "profileRetrievalStatus" : "RETRIEVED",
  "inventoryData" : {
   "subscriberHandsetSalesChannelCapableList" : {
    "shscc" : [ "YES" ]
   },
   "subscriberHomeSwitchTimeZoneList" : {
    "shstz" : [ "+1200" ]
   },
```



```
    “prepaidFlag”:true,
    "subscriberStatusList" : {
      "ss" : [ "ACTIVE" ]
     },
     "subscriberFullNamList" : {
      "sfn" : [ "Russel Crow" ]
     },
     "subscriberAccountTypeList" : {
      "sat" : [ "I" ]
     },
     "subscriberAccountSubTypeList" : {
      "sast" : [ "H" ]
     },
     "subscriberPrimaryMDNindList" : {
      "spmi" : [ "Yes" ]
     },
     "subscriberMDNList" : {
      "smdn" : [ "tel:+4433433312322" ]
     },
     "subscriberMEIDList" : {
      "smeid" : [ "000000000000" ]
     },
     "subscriberAccountAgeList" : {
      "saa" : [ "5" ]
     },
     "subscriberAccountList" : {
      "sa" : [ "222" ]
     }
   }
  } ]
}
```


Table 1: Success Scenarios and Canned Responses

Scenario	Steps	Result

Retrieve ACTIVE Subscriber Profile	Send a request with address tel: +15087300001.
	
	GET http://{host}:{port}/cxf/subscriberProfile/v1/sandbox?address=tel%3A%2B15087300001
		You will get a JSON response as shown in Sample 1 but with following difference.
		
		"subscriberStatusList" : {
		"ss" : ["ACTIVE"]
		}

Retrieve CANCELLED Subscriber Profile	Send a request with address tel: +15087300002.
	
	GET http://{host}:{port}/cxf/subscriberProfile/v1/sandbox?address=tel%3A%2B15087300002
		You will get a JSON response as shown in Sample 1 but with following difference.
		
		"subscriberStatusList" : {
		"ss" : ["CANCELLED"]
		}

Retrieve RESERVED Subscriber Profile	Send a request with address tel: +15087300003.
	
	GET http://{host}:{port}/cxf/subscriberProfile/v1/sandbox?address=tel%3A%2B15087300003
		You will get a JSON response as shown in Sample 1 but with following difference.
		
		"subscriberStatusList" : {
		"ss" : ["RESERVED"]
		}

Retrieve SUSPENDED Subscriber Profile	Send a request with address tel: +15087300004.
	
	GET http://{host}:{port}/cxf/subscriberProfile/v1/sandbox?address=tel%3A%2B15087300004
		You will get a JSON response as shown in Sample 1 but with following difference.
		
		"subscriberStatusList" : {
		"ss" : ["SUSPENDED"]
		}

Retrieve UNKNOWN Subscriber Profile	Send a request with a valid address not configured for a particular scenario with the format tel:+1XXXXXXXXXX.
	
	GET http://{host}:{port}/cxf/subscriberProfile/v1/sandbox?address=tel%3A%2B1XXXXXXXXXX
		You will get a JSON response as shown in Sample 1 but with following difference.
		
		"subscriberStatusList" : {
		"ss" : ["UNKNOWN"]
		}

A.2.2 Unsuccessful Scenarios

Service and Policy failure scenarios are handled as in a normal service as described in section 5, since the request comes through all components as normal before being diverted from reaching the real ESB endpoint.

End of Document

