



# RESTful Network API for Customer Profile

## Candidate Version 1.0 – 05 Mar 2013

---

**Open Mobile Alliance**  
OMA-TS-REST\_NetAPI\_CustomerProfile-V1\_0-20130305-C

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

**NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.**

**THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.**

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

# Contents

<b>1. SCOPE</b> .....	<b>5</b>
<b>2. REFERENCES</b> .....	<b>6</b>
<b>2.1 NORMATIVE REFERENCES</b> .....	<b>6</b>
<b>2.2 INFORMATIVE REFERENCES</b> .....	<b>6</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>7</b>
<b>3.1 CONVENTIONS</b> .....	<b>7</b>
<b>3.2 DEFINITIONS</b> .....	<b>7</b>
<b>3.3 ABBREVIATIONS</b> .....	<b>7</b>
<b>4. INTRODUCTION</b> .....	<b>8</b>
<b>4.1 VERSION 1.0</b> .....	<b>8</b>
<b>5. CUSTOMER PROFILE API DEFINITION</b> .....	<b>9</b>
<b>5.1 RESOURCES SUMMARY</b> .....	<b>9</b>
<b>5.2 DATA TYPES</b> .....	<b>12</b>
5.2.1 XML Namespaces.....	12
5.2.2 Structures .....	12
5.2.2.1 Type: AttributeNameList.....	12
5.2.2.2 Type: AttributeMetadata.....	12
5.2.2.3 Type: AttributeList.....	13
5.2.2.4 Type: Attribute.....	13
5.2.3 Enumerations .....	13
5.2.4 Values of the Link “rel” attribute.....	13
<b>5.3 SEQUENCE DIAGRAMS</b> .....	<b>13</b>
5.3.1 Retrieving profile metadata.....	13
5.3.2 Retrieving profile content .....	14
<b>6. DETAILED SPECIFICATION OF THE RESOURCES</b> .....	<b>16</b>
<b>6.1 RESOURCE: ATTRIBUTE NAME LIST</b> .....	<b>16</b>
6.1.1 Request URL variables .....	16
6.1.2 Response Codes and Error Handling .....	16
6.1.3 GET.....	17
6.1.3.1 Example: Retrieving a list of supported attribute names (Informative) .....	17
6.1.3.1.1 Request.....	17
6.1.3.1.2 Response.....	17
6.1.4 PUT.....	18
6.1.5 POST.....	18
6.1.6 DELETE .....	18
<b>6.2 RESOURCE: ATTRIBUTE LIST</b> .....	<b>18</b>
6.2.1 This resource is used for retrieving the attributes (all or an indicated selection). Request URL variables .....	18
6.2.2 Response Codes and Error Handling .....	18
6.2.3 GET.....	18
6.2.3.1 Example 1: Retrieving all supported attributes (some attribute names may not have an associated value) (Informative)..	20
6.2.3.1.1 Request.....	20
6.2.3.1.2 Response.....	20
6.2.3.2 Example 2: Retrieving selected attributes using profile name (Informative).....	21
6.2.3.2.1 Request.....	21
6.2.3.2.2 Response.....	21
6.2.3.3 Example 3: Retrieving a non-supported attribute (Informative).....	21
6.2.3.3.1 Request.....	21
6.2.3.3.2 Response.....	22
6.2.3.4 Example 4: Retrieving selected attributes using profile name with partial success (some selected attributes are not supported) (Informative).....	22
6.2.3.4.1 Request.....	22
6.2.3.4.2 Response.....	22
6.2.4 PUT.....	22

- 6.2.5 POST..... 23
- 6.2.6 DELETE ..... 23
- 7. FAULT DEFINITIONS ..... 24
  - 7.1 SERVICE EXCEPTIONS..... 24
  - 7.2 POLICY EXCEPTIONS ..... 24
- APPENDIX A. CHANGE HISTORY (INFORMATIVE)..... 25
  - A.1 APPROVED VERSION HISTORY ..... 25
  - A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY ..... 25
- APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ..... 27
  - B.1 SCR FOR REST.CP SERVER..... 27
    - B.1.1 SCR for REST.CP.AttributeNameList Server ..... 27
    - B.1.2 SCR for REST.CP.Attributes Server ..... 27
- APPENDIX C. APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE) ..... 28
- APPENDIX D. JSON EXAMPLES (INFORMATIVE) ..... 29
  - D.1 RETRIEVING A LIST OF SUPPORTED ATTRIBUTE NAMES (SECTION 6.1.3.1) ..... 29
  - D.2 RETRIEVING ALL SUPPORTED ATTRIBUTES (SOME ATTRIBUTE NAMES MAY NOT HAVE AN ASSOCIATED VALUE) (SECTION 6.2.3.1)..... 30
  - D.3 RETRIEVING SELECTED ATTRIBUTES USING PROFILE NAME (SECTION 6.2.3.2)..... 31
  - D.4 RETRIEVING A NON-EXISTING ATTRIBUTE (SECTION 6.2.3.3) ..... 31
  - D.5 RETRIEVING SELECTED ATTRIBUTES USING PROFILE NAME WITH PARTIAL SUCCESS (SOME SELECTED ATTRIBUTES ARE NOT SUPPORTED) (SECTION 6.2.3.4) ..... 32
- APPENDIX E. OPERATIONS MAPPING TO A PRE-EXISTING BASELINE SPECIFICATION (INFORMATIVE)..... 33
- APPENDIX F. LIGHT-WEIGHT RESOURCES (INFORMATIVE) ..... 34
- APPENDIX G. AUTHORIZATION ASPECTS (NORMATIVE) ..... 35
  - G.1 USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS..... 35
    - G.1.1 Scope values ..... 35
      - G.1.1.1 Definitions..... 35
      - G.1.1.2 Downscoping ..... 35
      - G.1.1.3 Mapping with resources and methods..... 36
    - G.1.2 Use of 'acr:auth' ..... 36
- APPENDIX H. ATTRIBUTE NAMES-VALUE PAIRS AND PROFILE NAMES (NORMATIVE) ..... 37

## Figures

- Figure 1 Resource structure defined by this specification..... 10
- Figure 2 Retrieve profile metadata ..... 14

## Tables

- Table 1: Scope values for RESTful Customer Profile API..... 35
- Table 2: Required scope values for: Querying of Customer Profile ..... 36

# 1. Scope

This specification defines a RESTful API for Customer Profile using HTTP protocol bindings.

## 2. References

### 2.1 Normative References

- [**Autho4API\_10**] “Authorization Framework for Network APIs”, Open Mobile Alliance™, OMA-ER-Autho4API-V1\_0, URL: <http://www.openmobilealliance.org/>
- [**E164**] “E164: The international public telecommunication numbering plan”, ITU, URL:<http://www.itu.int/rec/T-REC-E.164-201011-I/en>
- [**IETF\_ACR\_draft**] “The acr URI for anonymous users”, S.Jakobsson, K.Smith, January 2010, URL: <http://tools.ietf.org/html/draft-uri-acr-extension-00>
- [**ISO3166-1**] “ISO3166-1-alpha-2-code”, ISO, URL:[http://www.iso.org/iso/country\\_names\\_and\\_code\\_elements](http://www.iso.org/iso/country_names_and_code_elements)
- [**ISO639-1**] “ISO630-1:Codes for the representation of names of languages – Part 1:Alpha-2 code”, ISO, URL:[http://www.iso.org/iso/catalogue\\_detail?csnumber=22109](http://www.iso.org/iso/catalogue_detail?csnumber=22109)
- [**ISO8601-2004**] “Data elements and interchange formats – Information interchange – Representation of dates and times”, ISO, URL:[http://www.iso.org/iso/catalogue\\_detail?csnumber=40874](http://www.iso.org/iso/catalogue_detail?csnumber=40874)
- [**RD-REST\_NetAPI\_Customer Profile**] “OMA RESTful Network API for Customer Profile”, Open Mobile Alliance™, OMA-RD-REST\_NetAPI\_CustomerProfile-V1\_0, URL: <http://www.openmobilealliance.org/>
- [**REST\_NetAPI\_Common**] “Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST\_NetAPI\_Common-V1\_0, URL: <http://www.openmobilealliance.org/>
- [**REST\_SUP\_CustomerProfile**] “XML schema for the RESTful Network API for Customer Profile”, Open Mobile Alliance™, OMA-SUP-XSD\_rest\_netapi\_customerprofile-V1\_0, URL: <http://www.openmobilealliance.org/>
- [**RFC2119**] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [**RFC2616**] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL:<http://www.ietf.org/rfc/rfc2616.txt>
- [**RFC3261**] “SIP: Session Initiation Protocol”, J. Rosenberg et al., June 2002, URL: <http://www.rfc-editor.org/rfc/rfc3261.txt>
- [**RFC3966**] “The tel URI for Telephone Numbers”, H.Schulzrinne, December 2004, URL: <http://www.ietf.org/rfc/rfc3966.txt>
- [**RFC3986**] “Uniform Resource Identifier (URI): Generic Syntax”, R. Fielding et. al, January 2005, URL:<http://www.ietf.org/rfc/rfc3986.txt>
- [**RFC4627**] “The application/json Media Type for JavaScript Object Notation (JSON)”, D. Crockford, July 2006, URL: <http://www.ietf.org/rfc/rfc4627.txt>
- [**RFC5646**] “Tags for identifying languages”, A.Phillips et. al, September 2009, URL:<http://www.ietf.org/rfc/rfc5646.txt>
- [**SCRRULES**] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures, URL: <http://www.openmobilealliance.org/>
- [**XMLSchema1**] W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL: <http://www.w3.org/TR/xmlschema-1/>
- [**XMLSchema2**] W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL: <http://www.w3.org/TR/xmlschema-2/>

### 2.2 Informative References

- [**OMADICT**] “Dictionary for OMA Specifications”, Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2\_9, URL:<http://www.openmobilealliance.org/>
- [**REST\_WP**] “Guidelines for RESTful Network APIs”, Open Mobile Alliance™, OMA-WP-Guidelines\_for\_RESTful\_Network\_APIs, URL:<http://www.openmobilealliance.org/>

## 3. Terminology and Conventions

### 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

### 3.2 Definitions

**Customer Profile** See User Profile [OMADICT]

Additionally, all definitions from the OMA Dictionary apply [OMADICT].

### 3.3 Abbreviations

<b>ACR</b>	Anonymous Customer Reference
<b>API</b>	Application Programming Interface
<b>BCP</b>	Best Current Practices
<b>HTTP</b>	HyperText Transfer Protocol
<b>JSON</b>	JavaScript Object Notation
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>MSISDN</b>	Mobile Subscriber ISDN Number
<b>OMA</b>	Open Mobile Alliance
<b>REST</b>	REpresentational State Transfer
<b>SIP</b>	Session Initiation Protocol
<b>SCR</b>	Static Conformance Requirements
<b>SMS</b>	Short Message Service
<b>TS</b>	Technical Specification
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>WP</b>	White Paper
<b>XML</b>	eXtensible Markup Language
<b>XSD</b>	XML Schema Definition

## 4. Introduction

The Technical Specification of the RESTful Network API for Customer Profile contains HTTP protocol bindings based on the requirements for customer profile defined in [RD-REST\_NetAPI\_CustomerProfile], using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML and JSON).

### 4.1 Version 1.0

Version 1.0 of this specification supports the following operations:

- Retrieve customer profile metadata (i.e. attribute names)
- Retrieve all or selected attribute(s)

In addition, this specification provides:

- Support for scope values used with authorization framework defined in [Autho4API\_10]
- Support for Anonymous Customer Reference (ACR) as an end user identifier
- Support for “acr:auth” as a reserved keyword in a resource URL variable that identifies an end user



## 5. Customer Profile API definition

This section is organized to support a comprehensive understanding of the Customer Profile API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST\_NetAPI\_Common].

The remainder of this document is structured as follows:

Section 5 starts with a diagram representing the resources hierarchy followed by a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP methods, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. Application/x-www-form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D.

Section 7 contains fault definition details such as Service Exceptions and Policy Exceptions.

Appendix B provides the Static Conformance Requirements (SCR).

Appendix C provides application/x-www-form-urlencoded examples, where applicable.

Appendix E provides the operations mapping to a pre-existing baseline specification, where applicable.

Appendix F provides a list of all light-weight resources, where applicable.

Appendix G defines authorization aspects to control access to the resources defined in this specification.

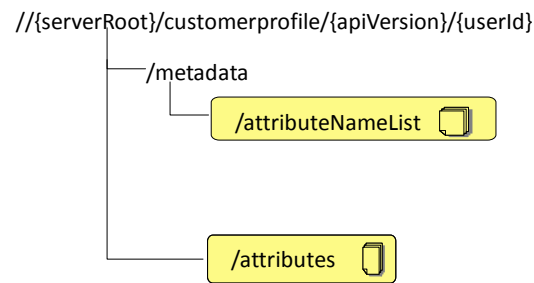
Appendix H defines attribute names/values and profile names for this specification.

Note: Throughout this document client and application can be used interchangeably.

### 5.1 Resources Summary

This section summarizes all the resources used by the RESTful Network API for Customer Profile.

The "apiVersion" URL variable SHALL have the value "v1" to indicate that the API corresponds to this version of the specification. See [REST\_NetAPI\_Common] which specifies the semantics of this variable.



**Figure 1 Resource structure defined by this specification**

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

**Purpose: To allow a client to retrieve profile information**

Resource	URL Base URL: http://{serverRoot}/customerprofile/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Attribute name list	{userId}/metadata/attributeNameList	AttributeNameList	This operation is used to retrieve a list of supported attribute names.	no	no	no
Attribute list	{userId}/attributes	AttributeList	This operation is used to retrieve attributes (filters are defined to limit the number of attributes).	no	no	no

## 5.2 Data Types

### 5.2.1 XML Namespaces

The XML namespace for the Customer Profile data types is:

urn:oma:xml:rest:netapi:customerprofile:1

The 'xsd' namespace prefix is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace prefix is used in the present document to refer to the data types defined in [REST\_NetAPI\_Common]. The use of namespace prefixes such as 'xsd' is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST\_SUP\_CustomerProfile].

### 5.2.2 Structures

The subsections of this section define the data structures used in the Customer Profile API.

Some of the structures can be instantiated as so-called root elements.

For structures that contain elements which describe a user identifier, the statements in section 6 regarding 'tel', 'sip' and 'acr' URI schemes apply.

This specification defines a number of attribute name-value pairs, each of whom MAY be associated with a profile name. Attribute names are retrievable as metadata, therefore allowing an application to find out all attribute name-value pairs that are supported (i.e. may be retrieved). Each attribute name-value pair MAY be associated with a uniquely identified profile name, which is also retrievable as part of the metadata. The grouping of attribute name-value pairs in a specific profile is targeted as a convenience for the application, by allowing the retrieval of the entire set of supported attribute name-value pairs that are included in the specified profile. See Appendix H for a list of defined attribute name-value pairs and associated profile names.

#### 5.2.2.1 Type: AttributeNameList

List of supported attribute names

Element	Type	Optional	Description
attributeMetadata	AttributeMetadata [1..unbounded]	No	A list of supported attribute metadata (attribute names and their corresponding profile). For a list of defined attribute names and values and their corresponding profiles, see Appendix H.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named attributeNameList of type AttributeNameList is allowed in response bodies.

#### 5.2.2.2 Type: AttributeMetadata

Attribute metadata

Element	Type	Optional	Description
attributeName	xsd:string	No	A supported attribute name. For a list of defined attribute names and values, see Appendix H.
profileName	xsd:string	Yes	A supported profile name, that includes the accompanying attribute name. For a list of defined attribute names and values and their corresponding profile, see Appendix H.

A root element named attributeMetadata of type AttributeMetadata is allowed in response bodies.

### 5.2.2.3 Type: AttributeList

Attributes retrievable as a list

Element	Type	Optional	Description
attribute	Attribute [1..unbounded]	No	Contains a list of attributes related to user. For a list of defined attribute names and values, see Appendix H.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named attributeList of type AttributeList is allowed in response bodies.

### 5.2.2.4 Type: Attribute

Individual attribute

Element	Type	Optional	Description
name	xsd:string	No	Name of the attribute. For a list of defined attribute names and values, see Appendix H.
value	xsd:string	Choice	Optional element; if present it provides the value of the attribute.
<any element>	< type is defined in a schema implementing the element>	Choice	Optional element; if present it provides the value of the attribute. Note that element 'any element' can be any element from any other namespace (schema) than the target namespace. Type of such element is defined by the schema implementing the element. In XML implementations, element "any" must be qualified with the namespace prefix.

XSD modelling uses an optional "choice" to select either a value or <any element>, or none of them.

## 5.2.3 Enumerations

This specification does not define any enumerations.

## 5.2.4 Values of the Link "rel" attribute

This specification does not define any elements of type Link.

## 5.3 Sequence Diagrams

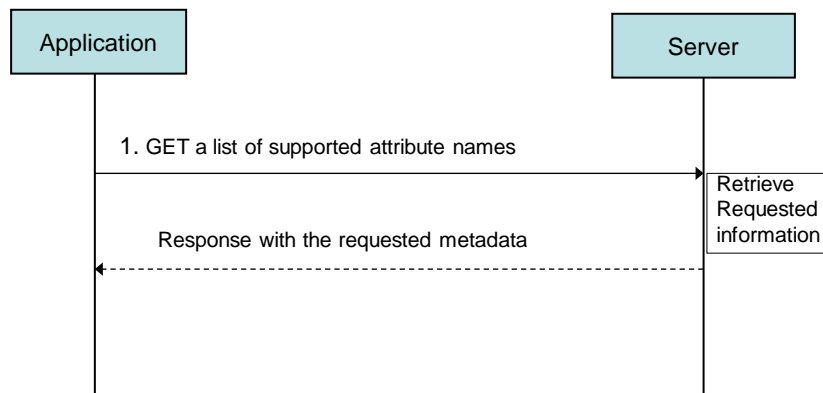
The following subsections describe the resources, methods and steps involved in typical scenarios.

### 5.3.1 Retrieving profile metadata

This figure below shows a scenario for retrieving profile metadata (i.e. a list of supported attribute names).

The resources:

- To retrieve a list of supported attribute names, read resource under **http://{serverRoot}/customerprofile/{apiVersion}/{userId}/metadata/attributeNameList**



**Figure 2 Retrieve profile metadata**

Outline of the flows:

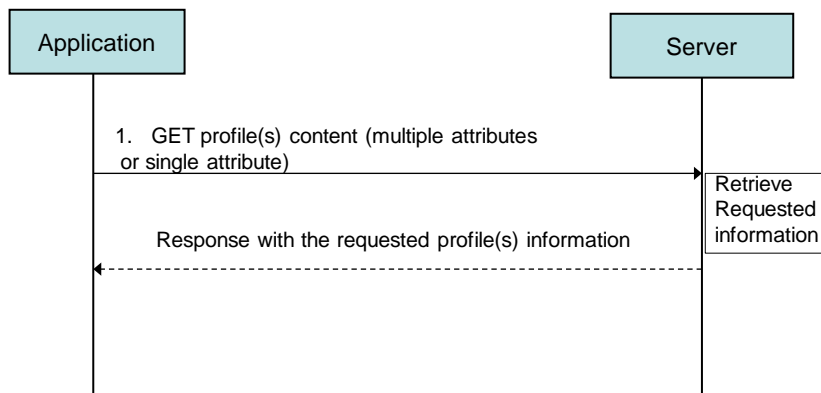
1. An application requests/receives profile metadata (i.e. a list of supported attribute names) for a user identified by `userId`, using GET.

### 5.3.2 Retrieving profile content

This figure below shows a scenario for retrieving profile content (i.e. all/selected attribute(s)).

The resources:

- To retrieve all/selected attributes, read resource under **`http://{serverRoot}/customerprofile/{apiVersion}/{userId}/attributes`**



**Figure 3 Retrieve profile content**

Outline of the flows:

1. An application requests/receives profile content (i.e. all/selected attribute(s)) for a user identified by `userId`, using GET.

## 6. Detailed specification of the resources

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML and JSON):

- Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) **MUST** be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in “resourceURL” and “link” elements).
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an MSISDN, it **MUST** be defined as a global number according to [RFC3966] (e.g. tel:+19585550100). The use of characters other than digits and the leading “+” sign **SHOULD** be avoided in order to ensure uniqueness of the resource URL. This applies regardless of whether the user identifier appears in a URL variable or in a parameter in the body of an HTTP message.
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of a SIP URI, it **MUST** be defined according to [RFC3261].
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it **MUST** be defined according to [IETF\_ACR\_draft], i.e. it **MUST** include the protocol prefix 'acr:' followed by the ACR.
  - The ACR ‘auth’ is a supported reserved keyword, and **MUST NOT** be assigned as an ACR to any particular end user. See G.1.2 for details regarding the use of this reserved keyword.
- For requests and responses that have a body, the following applies: in the requests received, the server **SHALL** support JSON and XML encoding of the parameters in the body. The Server **SHALL** return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST\_NetAPI\_Common]. In notifications to the Client, the server **SHALL** use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations **SHALL** follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST\_NetAPI\_Common].

### 6.1 Resource: Attribute name list

The resource used is:

**http://{serverRoot}/customerprofile/{apiVersion}/{userId}/metadata/attributeNameList**

This resource is used for retrieving profile metadata in the form of a list of attribute names.

#### 6.1.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
userId	user identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

#### 6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].



For Policy Exception and Service Exception fault codes applicable to the RESTful Customer Profile API, see section 7.

## 6.1.3 GET

This operation is used for retrieving a list of supported attribute names, and an optionally associated profile name, for a given user. The retrieved metadata represents the supported attribute name-value pairs for any user, and MAY include profile names associated to the attribute name-value pairs if such profile names have been specified (see Appendix H).

### 6.1.3.1 Example: Retrieving a list of supported attribute names (Informative)

#### 6.1.3.1.1 Request

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/metadata/attributeNameList HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cusprof:attributeNameList xmlns:cusprof="urn:oma:xml:rest:netapi:customerprofile:1">
  <attributeMetadata>
    <attributeName>country</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>locality</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>area</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>streetName</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>streetNumber</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>postalCode</attributeName>
    <profileName>addressProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>minAge18</attributeName>
    <profileName>verificationProfile</profileName>
  </attributeMetadata>
  <attributeMetadata>
    <attributeName>paymentType</attributeName>
```

```
<profileName>accountProfile</profileName>
</attributeMetadata>
```

```
<resourceURL>http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/metadata/attributeNameList</resourceURL>
</cusprof:attributeNameList>
```

## 6.1.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC2616].

## 6.1.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC2616].

## 6.1.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC2616].

## 6.2 Resource: Attribute list

The resource used is:

`http://{serverRoot}/customerprofile/{apiVersion}/{userId}/attributes`

### 6.2.1 This resource is used for retrieving the attributes (all or an indicated selection). Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1
userId	user identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

### 6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to the RESTful Customer Profile API, see section 7.

### 6.2.3 GET

This operation is used for retrieving the attributes from a given profile. The selection of attribute name-value pairs is determined using the query parameters “attrFilter” (whose value(s) are used to indicate individual attribute name value pair(s))

to be included in the response), and/or “profFilter” (whose value(s) are used to indicate entire profile name(s) (set(s) of attribute name value pairs to be included in the response).

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
attrFilter	xsd:string	Yes	<p>Defines selected attribute(s) to be returned in a GET response body.</p> <p>The syntax of this query parameter is: attrFilter=attrFilterValue, where attrFilterValue is an attribute name from the supported Attributed Name List. Any number of attrFilter=attrFilterValue pairs separated by &amp; are allowed.</p> <p>If attrFilterValue = attribute name (e.g. “givenName”), GET response body SHALL only include the selected attributes (i.e. “givenName” (see Appendix H)).</p> <p>If both attrFilter and profFilter are absent, GET response body SHALL include all the supported attributes,</p>
profFilter	xsd:string	Yes	<p>Defines selected sub-profile(s) to be returned in a GET response body.</p> <p>The syntax of this query parameter is: profFilter=profFilterValue, where profFilterValue is a profile name from the supported profile names (see Appendix H). Any number of profFilter=profFilterValue pairs separated by &amp; are allowed.</p> <p>If profFilterValue = sub-profile name (e.g. “svceAddressProfile”), GET response body SHALL include the attributes belonging to the defined profile (i.e. all attributes that belong to the profile “svceAddressProfile” (see Appendix H)).</p> <p>If both attrFilter and profFilter are absent, GET response body SHALL include all the supported attributes.</p> <p>If multiple profFilter query parameters are present, GET response body SHALL include the union of the sets of supported attributes belonging to the selected profiles. In addition, if attrFilter query parameters are also present, additionally selected attributes SHALL be included in the set of attributes returned.</p>

The selected set of attribute name-value pairs represents the union of the sets of supported attribute name value-pairs belonging to the selected profiles, to which any additionally selected attribute name-value pairs are added. The following cases are possible:

- The selected set of supported attribute name-value pairs is non-empty, and all selected attributes are supported. The response body SHALL contain all selected attribute name-value pairs (at least one attribute name-value pair, where the value MAY be empty, in which case the attribute name is returned, but the value is not returned). This represents a fully successful operation.
- The selected set of supported attribute name-value pairs is empty (i.e. the selection resulted in non-supported attribute name-value pairs). The response body SHALL contain a Service Exception (SVC002), and attribute list SHALL NOT be returned in the response body. This represents an unsuccessful operation.
- The selected set of supported attribute name-value pairs is non-empty, but not all selected attribute name-value pairs are supported (i.e. the attribute name-value pairs that can be returned are only a subset of the desired attribute name-value pairs). The response body SHALL contain all supported attribute name-value pairs matching the selection criteria (at least one attribute name-value pair, where the value MAY be empty, in which case the attribute name is returned, but the value is not returned), and no additional indication regarding the selected but not supported attribute name-value pairs SHALL be returned. This represents a partially successful operation.

### 6.2.3.1 Example 1: Retrieving all supported attributes (some attribute names may not have an associated value) (Informative)

#### 6.2.3.1.1 Request

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cusprof:attributeList xmlns:cusprof="urn:oma:xml:rest:netapi:customerprofile:1">
  <attribute>
    <name>country</name>
    <value>France</value>
  </attribute>
  <attribute>
    <name>locality</name>
    <value>Nice</value>
  </attribute>
  <attribute>
    <name>area</name>
  </attribute>
  <attribute>
    <name>streetName</name>
    <value>Rue des Jardins</value>
  </attribute>
  <attribute>
    <name>streetNumber</name>
    <value>1</value>
```

```

</attribute>
<attribute>
  <name>postalCode</name>
  <value>98765</value>
</attribute>
<attribute>
  <name>minAge18</name>
  <value>verifiedTrue</value>
</attribute>
<attribute>
  <name>paymentType</name>
  <value>prePaid</value>
</attribute>

<resourceURL>http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes</resourceURL>
</cusprof:attributeList>

```

### 6.2.3.2 Example 2: Retrieving selected attributes using profile name(Informative)

#### 6.2.3.2.1 Request

```

GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes?profFilter=accountProfile&attrFilter=postalCode HTTP/1.1
Accept: application/xml
Host: example.com

```

#### 6.2.3.2.2 Response

```

HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cusprof:attributeList xmlns:cusprof="urn:oma:xml:rest:netapi:customerprofile:1">
  <attribute>
    <name>paymentType</name>
    <value>prePaid</value>
  </attribute>
  <attribute>
    <name>postalCode</name>
    <value>98765</value>
  </attribute>
  <resourceURL>http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes</resourceURL>
</cusprof:attributeList>

```

### 6.2.3.3 Example 3: Retrieving a non-supported attribute (Informative)

#### 6.2.3.3.1 Request

```

GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes/attr_filter=birthDate HTTP/1.1
Accept: application/xml
Host: example.com

```

### 6.2.3.3.2 Response

HTTP/1.1 404 Not Found  
 Date: Thu, 24 Aug 2012 12:51:59 GMT  
 Content-Type: application/xml  
 Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC0002</messageId>
    <text>Invalid input value for message part %1</text>
    <variables>birthDate</variables>
  </serviceException>
</common:requestError>
```

### 6.2.3.4 Example 4: Retrieving selected attributes using profile name with partial success (some selected attributes are not supported) (Informative)

#### 6.2.3.4.1 Request

GET  
 /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes?profFilter=accountProfile&attrFilter=postalCode&attrFilter=telephoneHome  
 HTTP/1.1  
 Accept: application/xml  
 Host: example.com

#### 6.2.3.4.2 Response

HTTP/1.1 200 OK  
 Date: Thu, 24 Aug 2012 12:51:59 GMT  
 Content-Type: application/xml  
 Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<cusprof:attributeList xmlns:cusprof="urn:oma:xml:rest:netapi:customerprofile:1">
  <attribute>
    <name>paymentType</name>
    <value>prePaid</value>
  </attribute>
  <attribute>
    <name>postalCode</name>
    <value>98765</value>
  </attribute>
  <resourceURL>http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes</resourceURL>
</cusprof:attributeList>
```

## 6.2.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

## 6.2.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

## 6.2.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

## 7. Fault definitions

### 7.1 Service Exceptions

For common Service Exceptions refer to [REST\_NetAPI\_Common]. There are no additional Service Exception codes defined for the RESTful Customer Profile API.

### 7.2 Policy Exceptions

For common Policy Exceptions refer to [REST\_NetAPI\_Common]. There are no additional Service Exception codes defined for the RESTful Customer Profile API.



## Appendix A. Change History

(Informative)

### A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

### A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Version: REST_NetAPI_CustomerProfile-V1_0	13 Mar 2012	all	Initial baseline. Incorporates input to committee: OMA-ARC-2012-0062- INP_BaselineREST_NetAPI_CustomerProfileTS
	09 Aug 2012	many	OMA-ARC-REST-CusProf-2012-0010R01- INP_TS_initial_baseline_with_resource_hierarchy.doc
	22 Aug 2012	many	Implemented: OMA-ARC-REST-CusProf-2012-0011-CR_Version_section, OMA-ARC-REST-CusProf-2012-0012R02- CR_Resource_summary_section, OMA-ARC-REST-CusProf-2012-0013R01-CR_Structures_section
	07 Sep 2012	Section 5.1, 5.2, 5.3	Implemented: OMA-ARC-REST-CusProf-2012-0018R03- CR_Structures_additional, OMA-ARC-REST-CusProf-2012-0015- CR_Sequence_flows_section
	20 Sep 2012	Many	Implemented: OMA-ARC-REST-CusProf-2012-0016R02- CR_Detailed_spec_section, OMA-ARC-REST-CusProf-2012- 0017R01-CR_SCR_appendix
	17 Oct 20	Many	Implemented: OMA-ARC-REST-CusProf-2012-0024- CR_Correcting_XML_examples_and_adding_JSON
	29 Sep 2012	Appendix F	Implemented: OMA-ARC-REST-CusProf-2012-0026- CR_Appendix_F_lightweight_resources
	13 Nov 2012	Many	Implemented: OMA-ARC-REST-CusProf-2012-0028R02- CR_Section5_and_AppendixG_updates, OMA-ARC-REST-CusProf- 2012-0031-CR_Change_mandatory_SCR_to_optional
	12 Jan 2013	Many	Implemented: OMA-ARC-REST-CusProf-2012-0032- CR_Blueprint_for_auth_and_apply_template
	21 Jan 2013	Many	Implemented: OMA-ARC-REST-CusProf-2013-0004- CR_Addressing_majority_of_CONR_TS_comments
	12 Feb 2013	Many	Implemented: OMA-ARC-REST-CusProf-2013-0006- CR_Resolve_TS_CONR_comments_and_propose_simplify, editorials
	18 Feb 2013	Many	Implemented: OMA-ARC-REST-CusProf-2013-0010- CR_AttributeList_optionality_change, OMA-ARC-REST-CusProf- 2013-0011R01-CR_Addressing_TS_comments_after_simplification
	19 Feb 2013	Appendix B	Implemented: OMA-ARC-REST-CusProf-2013-0015-CR_TS_change_SCRs
	20 Feb 2013	Many	Implemented: OMA-ARC-REST-CusProf-2013-0012R01- CR_Appendix_H_attributes_for_discussion, OMA-ARC-REST- CusProf-2013-0016R01- CR_Handling_attribute_name_value_pairs_and_added_examples

Document Identifier	Date	Sections	Description
Candidate Version: REST_NetAPI_CustomerProfile- V1_0	05 Mar 2013	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2013-0073- INP_REST_NetAPI_CustomerProfile_V1.0_ERP_and_ETR_for_Ca ndidate_approval

## Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

### B.1 SCR for REST.CP Server

Item	Function	Reference	Requirement
REST-CP-SUPPORT-S-001-M	Support for the RESTful Customer Profile API	5,6	
REST-CP-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-CP-SUPPORT-S-003-M	Support for the JSON request & response format	6	

#### B.1.1 SCR for REST.CP.AttributeNameList Server

Item	Function	Reference	Requirement
REST-CP-ATTR-NAMELIST-S-001-O	Support list of attribute names	6.1	REST-CP-ATTR-NAMELIST-S-002-O
REST-CP-ATTR-NAMELIST-S-002-O	This operation retrieves all supported attribute names for a given profile - GET	6.1.3	

#### B.1.2 SCR for REST.CP.Attributes Server

Item	Function	Reference	Requirement
REST-CP-ATTRS-S-001-M	Support list of attributes	6.2	
REST-CP-ATTRS-S-002-M	This operation retrieves all or selected attributes of a given user, using optional “attrFilter” and/or “profFilter” - GET	6.2.3	

## Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations (Normative)

This specification does not define any API request based on application/x-www-form-urlencoded MIME type.

## Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a light-weight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC4627].

The following examples show the request and response for various operations using the JSON data format. The examples follow the XML to JSON serialization rules in [REST\_NetAPI\_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST\_NetAPI\_Common].

For full details on the operations themselves please refer to the section number indicated.

### D.1 Retrieving a list of supported attribute names (section 6.1.3.1)

Request:

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/metadata/attributeNameList HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attributeNameList": {
  "attributeMetadata": [
    {
      "attributeName": "country",
      "profileName": "addressProfile"
    },
    {
      "attributeName": "locality",
      "profileName": "addressProfile"
    },
    {
      "attributeName": "area",
      "profileName": "addressProfile"
    },
    {
      "attributeName": "streetName",
      "profileName": "addressProfile"
    },
    {
      "attributeName": "streetNumber",
      "profileName": "addressProfile"
    },
    {
      "attributeName": "postalCode",
      "profileName": "addressProfile"
    },
  ],
}
```

```
{
  "attributeName": "minAge18",
  "profileName": "verificationProfile"
},
{
  "attributeName": "paymentType",
  "profileName": "accountProfile"
}
],
"resourceURL": "http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/metadata/attributeNameList"
}}
```

## D.2 Retrieving all supported attributes (some attribute names may not have an associated value) (section 6.2.3.1)

### Request:

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes HTTP/1.1
Accept: application/json
Host: example.com
```

### Response:

```
HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn
```

```
{"attributeList": {
  "attribute": [
    {
      "name": "country",
      "value": "France"
    },
    {
      "name": "locality",
      "value": "Nice"
    },
    {
      "name": "area"
    },
    {
      "name": "streetName",
      "value": "Rue des Jardins"
    },
    {
      "name": "streetNumber",
      "value": "1"
    },
    {
      "name": "postalCode",
```

```

    "value": "98765"
  },
  {
    "name": "minAge18",
    "value": "verifiedTrue"
  },
  {
    "name": "paymentType",
    "value": "prePaid"
  }
],
"resourceURL": "http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes"
}}

```

### D.3 Retrieving selected attributes using profile name (section 6.2.3.2)

Request:

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes?profFilter=accountProfile&attrFilter=postalCode HTTP/1.1
```

Accept: application/json

Host: example.com

Response:

HTTP/1.1 200 OK

Date: Thu, 24 Aug 2012 12:51:59 GMT

Content-Type: application/json

Content-Length: nnnn

```

{"attributeList": {
  "attribute": [
    {
      "name": "paymentType",
      "value": "prePaid"
    },
    {
      "name": "postalCode",
      "value": "98765"
    }
  ]
},
"resourceURL": "http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes"
}}

```

### D.4 Retrieving a non-existing attribute (section 6.2.3.3)

Request:

```
GET /exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes/attr_filter=birthDate HTTP/1.1
Accept: application/json
Host: example.com
```

**Response:**

```
HTTP/1.1 404 Not Found
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"requestError": {"serviceException": {
  "messageId": "SVC0002",
  "text": "Invalid input value for message part %1",
  "variables": "birthDate"
}}}
```

## D.5 Retrieving selected attributes using profile name with partial success (some selected attributes are not supported) (section 6.2.3.4)

**Request:**

```
GET
/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes?profFilter=accountProfile&attrFilter=postalCode&attrFilter=telephoneHome
HTTP/1.1

Accept: application/json
Host: example.com
```

**Response:**

```
HTTP/1.1 200 OK
Date: Thu, 24 Aug 2012 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attributeList": {
  "attribute": [
    {
      "name": "paymentType",
      "value": "prePaid"
    },
    {
      "name": "postalCode",
      "value": "98765"
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/customerprofile/v1/tel%3A%2B19585550100/attributes"
}}
```



## Appendix E. Operations mapping to a pre-existing baseline specification (Informative)

As this specification does not have a baseline specification, this appendix is empty.

## Appendix F. Light-weight resources (Informative)

As this version of the specification does not define any light-weight resources, this appendix is empty.

## Appendix G. Authorization aspects (Normative)

This appendix specifies how to use the RESTful Customer Profile API in combination with some authorization frameworks.

### G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Customer Profile API MAY support the authorization framework defined in [Autho4API\_10].

A RESTful Customer Profile API supporting [Autho4API\_10]:

- SHALL conform to section D.1 of [REST\_NetAPI\_Common];
- SHALL conform to this section G.1.

#### G.1.1 Scope values

##### G.1.1.1 Definitions

In compliance with [Autho4API\_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Customer Profile API:

- SHALL support the scope values defined in the table below;
- MAY support scope values not defined in this specification.

Scope value	Description	For one-time access token
oma_rest_customerprofile.all_{apiVersion}	Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the “apiVersion” URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.	No
oma_rest_customerprofile.prof_{profileName}	Provide access for retrieving all attributes grouped in the specified profile.  The {profileName} part of the identifier SHALL have a value that matches one of the profiles defined in Appendix H.	No
oma_rest_customerprofile.attr_{attrName}	Provide access for retrieving the specified attribute.  The {attrName} part of the identifier SHALL have a value that exists in the list of supported attribute names.	No

Table 1: Scope values for RESTful Customer Profile API

##### G.1.1.2 Downscoping

In the case where the client requests authorization for “oma\_rest\_customerprofile.all\_{apiVersion}” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma\_rest\_customerprofile.prof\_{profileName}”
- “oma\_rest\_customerprofile.attr\_{attrName}”

### G.1.1.3 Mapping with resources and methods

Tables in this section specify how the scope values defined in section G.1.1.1 for the RESTful Customer Profile API map to the REST resources and methods of this API. In these tables, the root “oma\_rest\_customerprofile.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/customerprofile/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Attribute name list	{userId}/metadata/attributeNameList	6.2	n/a (see note below)	no	no	no
Attribute list	{userId}/attributes	6.4	<b>all_{apiVersion} or oma_rest_customerprofile.attr_{attrName} or oma_rest_customerprofile.prof_{profileName}</b>	no	no	no

**Table 2: Required scope values for: Querying of Customer Profile**

Note: no scope values are defined to control access to the resources that allow retrieving of metadata (i.e. “Attribute name list”). Service Providers may limit or refuse access to those resources through means that are out-of-scope for this specification. In the absence of Service Provider policies that restrict access to those resources (e.g. only known application identifiers are allowed), any application is able to retrieve the list of attribute names for a given profile.

### G.1.2 Use of ‘acr:auth’

This section specifies the use of ‘acr:auth’ in place of an end user identifier in a resource URL path.

An ‘acr’ URI of the form ‘acr:auth’, where ‘auth’ is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

A client MAY use ‘acr:auth’ in a resource URL in place of the {userId} resource URL variable in the resource URL path, when the RESTful Customer Profile API is used in combination with [Autho4API\_10].

In the case the RESTful Customer Profile API supports [Autho4API\_10], the server:

- SHALL accept ‘acr:auth’ as a valid value for the resource URL variable {userId}.
- SHALL conform to [REST\_NetAPI\_Common] section 5.8.1.1 regarding the processing of ‘acr:auth’.

## Appendix H. Attribute names-value pairs and profile names(Normative)

This specification defines on purpose a limited number of attribute name-value pairs and associated profile names that SHOULD commonly be supported by implementations but could be limited by service provider's policies . In addition, implementations MAY support additional attribute name-value pairs.

The following recommended attribute name-value pairs and their corresponding profiles are defined in this specification. A "profFilterValue" used to indicate a selection of attributes to be returned ((section 6.2.3) MUST match one of the profile names defined in this specifications, or added by an implementation. A "profileName" in "oma\_rest\_customerprofile.prof\_{profileName}" scope value (Appendix G) MUST match one of the profile names defined in this specifications, or added by an implementation.

Name	Description	Values	Profile names
country	The country portion of the user's address	Free text (e.g. France)	addressProfile
region	The region (e.g. state, province) portion of the user's address	Free text (e.g.Provence)	addressProfile
locality	The locality portion of the user's address (e.g. village, town, city)	Free text (e.g.Nice)	addressProfile
area	The subdivision portion of the user's address (e.g. neighbourhood, suburb, district)	Free text (e.g.Beaubanlieu)	addressProfile
streetName	The street portion of the user's address	Free text (e.g.Rue des Jardins)	addressProfile
streetNumber	The house number portion of the user's address	Free text (e.g. 1)	addressProfile
aptNumber	The apartment number portion of the user's address	Free text (e.g. apt.17A)	addressProfile
postalCode	The code for postal delivery portion of the user's address (e.g. ZIP code)	Free text (e.g.98765)	addressProfile
addressExtension	Additional data needed to indicate e.g. "attention to" or "c/o".	Free text (e.g. c/o John Doe)	addressProfile
name	End user's full name in displayable form including all name parts, ordered according to end user's locale and preferences	Free text (e.g. Mr Jean Alain Doe Jr)	nameProfile
title	End user's title	Free text (e.g: Mrs, Ms, Miss, Mr, Mstr, Dr, Ing, Esq, Mme, Mlle)	nameProfile
givenName	End user's given name or first name	Free text (e.g. Jean)	nameProfile
familyName	End user's surname, name or lastname	Free Text (e.g.Doe)	nameProfile
middleName	End user's middle name	Free text (e.g. Alain)	nameProfile
suffix	End user's suffix or generation identifier	Free text (e.g. Jr., Sr., III)	nameProfile

displayName	End user's suggested name to display in user interfaces (e.g. in an instant messaging buddy list)	Free text (e.g. Johnny)	nameProfile
telephoneHome	End user's home telephone number	E.164 [E.164] format +<CountryCode><City/AreaCode><LocalNumber> is recommended.  E.g. +1 (425) 555-1212).	contactProfile
mobileHome	End user's mobile telephone number	E.164 [E.164] format +<CountryCode><City/AreaCode><LocalNumber> is recommended.  E.g. +1 (425) 555-1212)	contactProfile
emailHome	End user's home email address	E.g. mailto:jean.doe@isp.com	contactProfile
telephoneWork	End user's work telephone number	E.164 [E.164] format +<CountryCode><City/AreaCode><LocalNumber>;ext=<ext> is recommended.  E.g. +1 (425) 555-1313; ext=123	workContactProfile
mobileWork	End user's work mobile telephone number	E.164 [E.164] format +<CountryCode><City/AreaCode><LocalNumber> is recommended.  E.g. +1 (425) 555-1212)	workContactProfile
emailWork	End user's work email address	E.g. mailto:jean.doe@company.com	workContactProfile
monthlyDataQuota	End user's monthly data plan quota	Number of megabytes, where 1 megabyte is 1048576 bytes (e.g.2000)	serviceProfile

monthlyVoiceQuota	End user's monthly voice plan quota	Number of minutes (e.g. 500)	serviceProfile
monthlySmsQuota	End user's monthly SMS plan quota	Number of messages (e.g. 200)	serviceProfile
dataQuotaRemaining	End user's data quota remaining	Number of megabytes, where 1 megabyte is 1048576 bytes (e.g.1000)	serviceProfile
voiceQuotaRemaining	End user's voice quota remaining	Number of minutes (e.g. 250)	serviceProfile
smsQuotaRemaining	End user's SMS quota remaining	Number of messages (e.g. 100)	serviceProfile
pictureURL	URL of the end user's picture	RFC3986 [RFC3986] URL format is recommended. E.g. URL:http://www.example.com/pictures/jeandoe	webProfile
websiteURL	URL of the end user's web page or blog	RFC3986 [RFC3986] URL format is recommended. E.g. URL:http://www.example.com/webpage/jeandoe	webProfile
age	End user's age	Age in years (e.g. 100)	personalProfile
birthDate	End user's birthdate	End user's birth date, represented as an ISO 8601:2004 [ISO8601-2004] YYYY-MM-DD	personalProfile
gender	End user's gender	Supported values are: female, male. Other values MAY be used when neither of the defined values are applicable.	personalProfile

locale	End user's locale	BCP47 [RFC5646] language tag format is recommended. This is typically an ISO 639-1 Alpha-2 [ISO639-1] language code in lower case and an ISO 3166-1 Alpha-2 [ISO3166-1] country code in uppercase, separated by dash (e.g. en-US, fr-CA). Implementations MAY choose to accept the use of “_” instead of “-”.	preferenceProfile
paymentType	An indication of the end user's payment preference	Supported values are: prePaid, postPaid, Other values (e.g. smartLimit) MAY be used when neither of the defined values are applicable.	accountProfile
accountStatus	Account status	Supported values are: active, suspended, unknown. Other values MAY be used when neither of the defined values are applicable.	accountProfile
minAge18	An indication of whether the end user is at least of age 18. If additional age specific attribute names for age verification are necessary to be specified, they MUST follow this pattern. The age MUST not contain any leading 0 digits.	Supported values are: verifiedTrue, verifiedFalse, unverified.	verificationProfile