RESTful Network API for Address Book

Candidate Version 1.0 – 07 Feb 2012

Open Mobile Alliance

OMA-TS-REST_NetAPI_AddressBook-V1_0-20120207-C

Contents

Figures

Tables

1. Scope

This specification defines a RESTful Address Book API using an HTTP protocol binding, based on the similar API defined in [3GPP 29.199-13].

2. References

2.1 Normative References

[3GPP 29.199-13] 3GPP Technical Specification, "Open Service Access (OSA); Parlay X Web Services; Part 18: Address List Management (Release 8)", URL:http://www.3gpp.org/

[Autho4API_10] "Authorization Framework for Network APIs", Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL: http://www.openmobilealliance.org/

[IETF_ACR_draft] "The acr URI for anonymous users", S.Jakobsson, K.Smith, July 2011, URL: http://tools.ietf.org/html/draft-uri-acr-extension-03

[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_NetAPI_Notification Channel] "RESTful Network API for Notification Channel", Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, URL: http://www.openmobilealliance.org/

[REST_SUP_AddressBook] "XML schema for the RESTful Network API for Address Book", Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_addressbook-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

[RFC2426] "vCard MIME Directory Profile", F.Dawson et al,. September 1998, URL:http://www.ietf.org/rfc/rfc2426.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", T. Berners-Lee, R. Fielding, L. Masinter, 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

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[W3C_URLENC] HTML 4.01 Specification, Section 17.13.4 Form content types, The World Wide Web Consortium, URL: http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1

[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_8, URL:http://www.openmobilealliance.org/

[ParlayREST_AddressListMgmt] "RESTful bindings for Parlay X Web Services – Address List Management", Version 1.0, Open Mobile Alliance™, OMA-TS-ParlayREST_AddressListManagement-V1_0

[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

For the purpose of this TS, all definitions from the OMA Dictionary apply [OMADICT].

Client-side Notification URL	An HTTP URL exposed by a client, on which it is capable of receiving notifications and that can be used by the client when subscribing to notifications.
Long Polling	A variation of the traditional polling technique, where the server does not reply to a request unless a particular event, status or timeout has occurred. Once the server has sent a response, it closes the connection, and typically the client immediately sends a new request. This allows the emulation of an information push from a server to a client.
Notification Channel	A channel created on the request of the client and used to deliver notifications from a server to an application. The channel is represented as a resource and provides means for the server to post notifications and for the client to receive them via specified delivery mechanisms. For example in the case of Long Polling the channel resource is defined by a pair of URLs. One of the URLs is used by the client as a call-back URL when subscribing for notifications. The other URL is used by the client to retrieve notifications from the Notification Server.
Notification Server	A server that is capable of creating and maintaining Notification Channels.
Server-side Notification URL	An HTTP URL exposed by a Notification Server, that identifies a Notification Channel and that can be used by a client when subscribing to notifications.

3.3 Abbreviations

AB	Address Book
ACR	Anonymous Customer Reference
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
OMA	Open Mobile Alliance
MIME	Multipurpose Internet Mail Extensions
REST	REpresentational State Transfer
SCR	Static Conformance Requirements
SIP	Session Initiation Protocol
TS	Technical Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

VCard	Virtual-Information Card
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The Technical Specification for the RESTful Network API for Address Book (AB) contains the HTTP protocol binding based on the Parlay X Address List Management Web Services [3GPP 29.199-13] specification, 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, JSON, and application/x-www-form-urlencoded).

4.1 Version 1.0

The RESTful Network API for Address Book V1.0 is a republication of the ParlayREST Address List Management API V1.0 [ParlayREST_AddressListMgmt] as part of the suite of OMA RESTful Network APIs.
Bug fixes and structural changes to fit that suite, but also functional changes have been applied.

Version 1.0 of the RESTful Network API for Address Book keeps supporting the following operations:

- Manage attributes related to a list
- Manage attributes related to a contact or a member in a list
- Manage contacts
- Manage lists
- Manage members in a list
- Manage nested lists related to a list
- Manage shared contacts or lists
- Manage subscriptions to contact or list changes
- Send notifications about contact or list changes

The following new functionality has been introduced:

- Transfer a member from one list to another
- Support “vCard 2.1” and “vCard 3.0” as contact attributes [RFC2426]
- 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:Authorization” as a reserved keyword in a resource URL variable that identifies an end user
- Resources, data types and element names have been changed to allow future extensions and reflect the API re-naming from Address List Management to Address Book

5. Address Book API definition

This section is organized to support a comprehensive understanding of the RESTful Address Book 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 **Error! Reference source not found.**). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains the detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP commands, the possible HTTP response codes, 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. 0 provides the Static Conformance Requirements (SCR).

Appendix E lists the Parlay X equivalent method for each supported REST resource and method combination, 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.

Finally, Appendix A illustrates the most common data structures and their relationships in Address Book.

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 Address Book API.

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.

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.

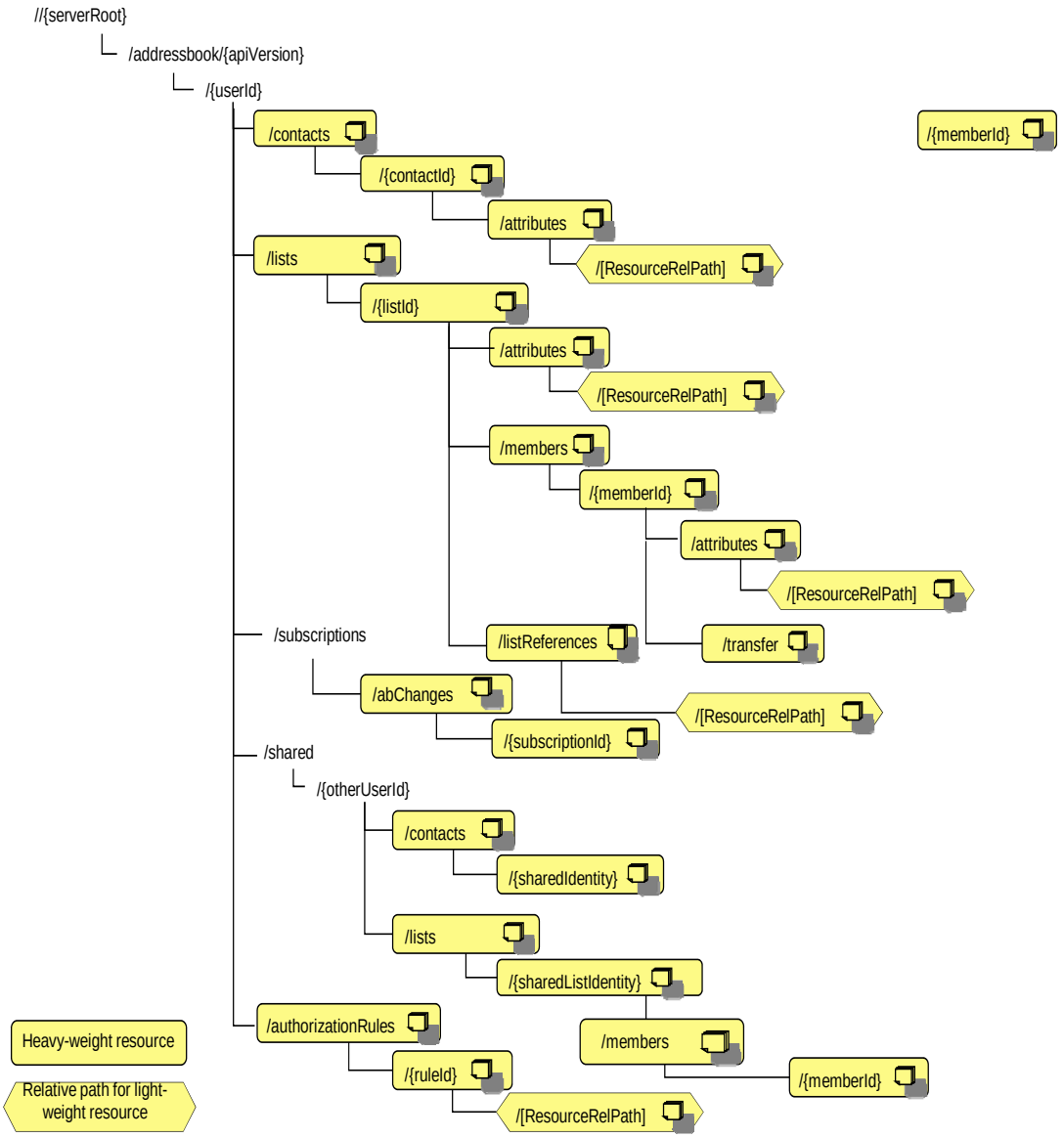

Figure 1 Resource structure defined by this specification

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 client to manage all contacts

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of contacts	/{userId}/contacts	ContactCollection	This operation retrieves information about all user's contacts.	no	no	no
Individual contact	/{userId}/contacts/{contactId}	Contact	This operation retrieves information about individual contact from the collection of user's contacts.	This operation creates a new contact or updates information about an existing contact.	no	This operation removes a contact.
Attributes for a contact	/{userId}/contacts/{contactId}/attributes	AttributeList	This operation returns all attributes for a contact.	no	no	no
Individual attribute for a contact	/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]	Attribute	This operation returns the value of the attribute for a contact.	This operation creates or updates an attribute for contact.	no	This operation removes an attribute for a contact.

Purpose: To allow client to manage its lists and members in lists

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Lists	/{userId}/lists	ListCollection	This operation retrieves all lists the user has created.	no	no	no
Individual list	/{userId}/lists/{listId}	List	This operation retrieves the list.	This operation creates or updates the entire list.	no	This operation removes the list from the system including attributes and members.
Attributes for a list	/{userId}/lists/{listId}/attributes	AttributeList	This operation returns all attributes for a list.	no	no	no
Individual attribute for a list	/{userId}/lists/{listId}/attributes/[ResourceRelPath]	Attribute	This operation returns the value of the list attribute.	This operation creates or updates an attribute	no	This operation deletes an attribute
Members in a list	/{userId}/lists/{listId}/members	MemberCollection	This operation retrieves the members in the list.	no	no	no
Individual member in a list	/{userId}/lists/{listId}/members/{memberId}	Member	This operation retrieves a user from a list. (Normally only used to verify the existence of the member)	This operation creates and updates a member of the list.	no	This operation removes the member from the list.
Attributes for a member in a list	/{userId}/lists/{listId}/members/{memberId}/attributes	AttributeList	This operation returns all attributes for a member.	no	no	no
Individual attribute for a member in a	/{userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]	Attribute	This operation returns the value of the attribute for a member.	This operation creates or updates an attribute for a	no	This operation removes an attribute for a

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
list				member.		member.
Individual member transfer	/{userId}/lists/{listId}/members/{memberId}/transfer	MemberTransferParameters (used for POST request) common:ResourceReference (used for POST response)	no	no	This operation transfers a member from one list to another list	no
List references	/{userId}/lists/{listId}/listReferences	ListReferenceCollection	This operation returns a list of references to other lists.	no	no	no
Individual list reference	/{userId}/lists/{listId}/listReferences/[ResourceRelPath]	common:Link	This operation returns the value of list reference.	This operation creates a new list reference.	no	This operation removes a list reference.

Purpose: To allow client to manage subscriptions to address book changes

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Address Book changes subscriptions	/{userId}/subscriptions/abChanges	AbChangesSubscriptionCollection	This operation retrieves all active subscriptions for address book changes.	no	This operation creates a new subscription for address book changes.	no
Individual address book changes subscription	/{userId/ subscriptions/ abChanges/{subscriptionId}	AbChangesSubscription	This operation retrieves information about individual subscription.	This operation updates/extends the duration of the subscription.	no	This operation deletes and terminates subscription.

Purpose: To allow client to receive notifications about address book changes

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Client resource for address book changes notifications	<specified by client in the subscription>	AbChangesNotification	no	no	This operation notifies a clientabout address book changes	no

Purpose: To allow client access to the contacts shared by another user

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of shared contacts	/{userId}/shared/{otherUserId}/contacts	ContactCollection	This operation retrieves all contacts shared by another user identified by {otherUserId} with the requsting user identified by {userId}.	no	no	no
Individual shared contact	/{userId}/shared/{otherUserId}/contacts/{sharedIdentity}	Contact	This operation retrieves information about individual contact shared by user identified by {otherUserId} with requesting user identified by {userId}.	no	no	no

Purpose: To allow client access to the lists shared by another user

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of shared lists	/{userId}/shared/{otherUserId}/lists	ListCollection	This operation retrieves all lists shared by another user identified by {otherUserId} with the requsting user identified by {userId}.	no	no	no
Individual shared list	/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}	List	This operation retrieves information about individual list shared by user identified by {otherUserId} with requesting user identified by {userId}.	no	no	no
Members in a shared list	/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}/members	MemberCollection	This operation retrieves the members in the list shared by user identified by {otherUserId} with requesting user identified by {userId}.	no	no	no
Individual member information from shared list	/{userId}/shared/lists/{otherUserId}/{sharedListIdentity}/members/{memberId}	Member	This operation retrieves individual member information from the shared list.	no	no	no

Purpose: To allow client access to authorization rules for shared contacts and lists

Resource	Base URL: http://{serverRoot}/addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Authorization Rules	/{userId}/authorizationRules	RuleList (Used for GET) Rule (Used for POST) common:ResourceReference (optional alternative for POST response)	This operation retrieves all authorization rules	no	This operation creates new authorization rule	no
Individual authorization rule	/{userId}/authorizationRules/{ruleId}	Rule (Used for PUT/GET)	This operation retrieves individual authorization rule	This operation updates individual authorization rule	no	This operation removes individual authorization rule
Individual authorization rule data	/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]	The data structure corresponds to the element pointed out by the request-URL. (Used for PUT/GET)	This operation retrieves data as specified in the URL	This operation creates or updates data as specified in the URL	no	This operation removes the data as specified in the URL

5.2 Data Types

5.2.1 XML Namespaces

The namespace for the Address Book data types is:

urn:oma:xml:rest:netapi:addressbook:1

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace is used in the present document to refer to the data types defined in [REST_NetAPI_Common]. The use of the names 'xsd' and ‘common’ is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST_SUP_AddressBook].

Applications following the RESTful Network API for Address Book V 1.0 specification SHALL use the namespace urn:oma:xml:rest:netapi:addressbook:1.

5.2.2 Structures

The subsections of this section define the data structures used in the RESTful Address Book API.

Some of the structures can be instantiated as so-called root elements, i.e. they define the type of a representation of a so-called heavy-weight resource.

The column [ResourceRelPath] in the tables below, if used, includes relative resource paths for light-weight resource URLs that are used to access individual elements in the data structure (so-called light-weight resources). A string from this column needs to be appended to the corresponding heavy-weight resource URL in order to create light-weight resource URL for that particular element in the data structure. “Not applicable” means that individual access to that element is not supported. The root element and data type of the resource associated with the [ResourceRelPath] are defined by the Element and Type columns in the row that defines the [ResourceRelPath].

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

5.2.2.1 Type: ContactCollection

Collection of all individual contacts for a user, in the form of a flat unique list

Element	Type	Optional	Description
contact	Contact [0..unbounded]	Yes	Contains a list of user’s contacts
resourceURL	xsd:anyURI	No	Self referring URL

A root element named contactCollection of type ContactCollection is allowed in response bodies.

5.2.2.2 Type: Contact

Individual contact information in the collection of contacts

Element	Type	Optional	Description
contactId	xsd:anyURI	No	Contains a relative URI of the contact (e.g. “Alice”). It is a key property of the element and SHALL NOT be altered when included in the light-weight resource URL. If contactId is also part of the request URL, the two MUST have the same value.
sharedIdentity	SharedIdentity	Yes	Contains a list of publicly known identities (absolute URIs) of the contact. A shared identity MAY be used by a requesting user to identify the contact in addition to the contactId (since the contactId may not be known by others).
attributeList	AttributeList	Yes	Contains a list of attributes (e.g. “display-name”) related to a contact.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link [0..unbounded]	Yes	MAY contain references to members (rel=”Member”) in type List associated with the contact. When a member containing a link to a contact is added to a list (or updated), a reverse link SHALL be created automatically by the server pointing from the contact to the actual member. When a member containing a link to a contact is removed from a list (or member is updated and link removed), the reverse link pointing from the contact to the member SHALL be removed automatically by the server. Links SHALL NOT be included in responses for a shared contact, unless the member referred by the link is a member in a list shared with the requester.

A root element named contact of type Contact is allowed in request and/or response bodies.

5.2.2.3 Type: SharedIdentity

List of publicly known identities

Element	Type	Optional	Description
sharedId	xsd:anyURI [0..unbounded]	Yes	Contains a list of absolute URIs (e.g. 'sip' URI, 'tel' URI, 'acr' URI) representing contact identities or list identities known to other users, used when retrieving a contact, respectively alist.

5.2.2.4 Type: ListCollection

List of lists

Element	Type	Optional	Description
list	List [0..unbounded]	Yes	Contains a list of list identities.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named listCollection of type ListCollection is allowed in response bodies.

5.2.2.5 Type: List

List of identities

Element	Type	Optional	Description
listId	xsd:anyURI	No	Contains a relative URI representing the list identity (e.g. “friends”). It is a key property of the element and SHALL NOT be altered when included in the light-weight resource URL. If listId is also part of the request URL, the two MUST have the same value.
memberCollection	MemberCollection	Yes	Members in the list
listReferenceCollection	ListReferenceCollection	Yes	Contains references to other lists.
category	ListType [0..unbounded]	Yes	Specifies the type of list. URIList is default if not specified.
sharedListIdentity	SharedIdentity	Yes	Contains a list of publicly known identities (absolute URIs) of the list (e.g. mailto:friendslist@example.com). A shared identity MAY be used by a requesting user to identify the list in addition to the listId (since the listId may not be known by others).
attributeList	AttributeList	Yes	Contains a list of attributes related to a list.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named list of type List is allowed in response bodies.

5.2.2.6 Type: AttributeList

Attributes of a list

Element	Type	Optional	[ResourceRelPath]	Description
attribute	Attribute [0..unbounded]	Yes	{name}	Contains a list of attributes related to contact, list or a member. The sub-element ‘name’ of the type Attribute SHALL NOT be altered when this element is accessed as a light-weight resource.
resourceURL	xsd:anyURI	No	Not applicable	Self referring URL

A root element named attributeList of type AttributeList is allowed in request and/or response bodies.

Column [ResourceRelPath] includes relative resource paths for light-weight resource URLs that are used to access individual elements in the data structure. A string from this column needs to be appended to the corresponding heavy-weight resource URL in order to create light-weight resource URL for that particular element in the data structure. “Not applicable” means that individual access to that element is not supported. The root element and data type of the resource associated with the [ResourceRelPath] are defined by the Element and Type columns in the row that defines the [ResourceRelPath].

5.2.2.7 Type: Attribute

Individual attribute of a list

Element	Type	Optional	Description
name	xsd:string	No	Name of the attribute. The following names are predefined by this specification: “display-name”. It is a key property of the element and SHALL NOT be altered when included in the light-weight resource URL. “vCard2.1” and “vCard3.0” are reserved names, and SHALL only be used to access the entire vCard as an opaque object [RFC2426]. When ‘name’ is “vCard2.1” or “vCard3.0”, it MUST be associated with the ‘objectValue’ choice to access an entire opaque vCard2.1 or vCard3.0, respectively.
value	xsd:string	Choice	Optional element; if present it provides the value of the attribute.
objectValue	xsd:base64Binary	Choice	Optional element; if present it contains a base64Binary object. In case name is “vCard2.1” this element SHALL contain a base64Binary encoded vCard2.1 object. In case name is “vCard3.0” this element SHALL contain a base64Binary encoded vCard3.0 object.
<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 an objectValue or <any element>, or none of them.

A root element named attribute of type Attribute is allowed in request and/or response bodies.

5.2.2.8 Type: MemberCollection

Collection of members that are part of a list

Element	Type	Optional	Description
member	Member [0..unbounded]	Yes	Contains a list of members related to a members list.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named memberCollection of type MemberCollection is allowed in response bodies.

5.2.2.9 Type: Member

Individual member information in a list

Element	Type	Optional	Description
memberId	xsd:anyURI	No	Contains an identifier of a single member (e.g. 'sip' URI, 'tel' URI, 'acr' URI). It is a key property of the element and SHALL NOT be altered when included in the light-weight resource URL. If memberId is also part of the request URL, the two MUST have the same value.
attributeList	AttributeList	Yes	Contains a list of attributes (e.g. display name) related to a member.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link [0..unbounded]	Yes	MAY contain references to contacts (rel="Contact") in type 'ContactCollection' associated with the member. When a contact containing a link to a member is added to the 'ContactCollection' (or updated), a reverse link SHALL be created automatically by the server pointing from the member to the actual contact. When a contact containing a link to a member is removed from the 'ContactCollection' (or contact is updated and link removed), the reverse link pointing from the member to the contact SHALL be removed automatically by the server. Links SHALL NOT be included in responses for a shared list, unless the contact referred by the link is also shared with the requester.

A root element named member of type Member is allowed in request and/or response bodies.

5.2.2.10 Type: ListReferenceCollection

List of references to other lists

Element	Type	Optional	[ResourceRelPath]	Description
resourceURL	xsd:anyURI	No	Not applicable	Self referring URL
link	common:Link [0..unbounded]	Yes	{href}	Contains references to other lists (rel="List").

A root element named listReferenceCollection of type ListReferenceCollection is allowed in response bodies.

5.2.2.11 Type: AbChangesSubscriptionCollection

Collection of subscriptions to notifications of changes in contacts and/or lists

Element	Type	Optional	Description
abChangesSubscription	AbChangesSubscription [0..unbounded]	Yes	Contains a list of subscriptions related to contacts changes and/or list changes.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named abChangesSubscriptionCollection of type AbChangesSubscriptionCollection is allowed in response bodies.

5.2.2.12 Type: AbChangesSubscription

Individual subscription to notifications of changes in contacts and/or a list

Element	Type	Optional	Description
anyContacts	(empty)	Choice	Indicates a subscription for changes about any contacts in the 'ContactCollection'.
listId	xsd:anyURI	Choice	Indicates a subscription for changes about the specified list.
callbackReference	common:CallbackReference	No	Client's notification URL and OPTIONAL callbackData
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
duration	xsd:int	Yes	Subscription duration in seconds. Server would expire subscription after specified number of seconds after subscription creation. If not specified – default value assigned by the server.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.

XSD modelling uses a "choice" to select either anyContacts or listId.
A root element named abChangesSubscription of type AbChangesSubscription is allowed in request and response bodies.

Note that the clientCorrelator is used for purposes of error recovery as specified in [REST_NetAPI_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. The document [REST_NetAPI_Common] provides a recommendation regarding the generation of the value of this field.

5.2.2.13 Type: AbChangesNotification

Notifications of changes in contacts and/or a list

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating a subscription to notifications of changes in contacts and/or a list. See [REST_NetAPI_Common] for details.
resourceStatus	ResourceStatus	No	Indicates the state for the subscription.
duration	xsd:int	Yes	Represents the time in seconds that the subscription is active, starting from the time the subscription is created by the server.
link	common:Link [1..unbounded]	No	SHALL contain reference to the added or updated member or contact (rel="Member" or rel="Contact" respectively). A notification MAY include several links for each member or contact. SHALL contain reference to the list in case a member or contact has been removed (rel="List" or rel="ContactCollection" respectively). SHALL contain reference to the subscription (rel="AbChangesSubscription") if resourceStatus is 'Active'.

A root element named abChangesNotification of type AbChangesNotification is used by server in the POST request to the client informing it about change in the list that client is monitoring (using one of the subscriptions).

5.2.2.14 Type: RuleList

List of authorization rules

Element	Type	Optional	Description
rule	Rule [0..unbounded]	Yes	Contains a list of all authorization rules related to contacts and lists.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named ruleList of type RuleList is allowed in response bodies.

5.2.2.15 Type: Rule

Individual authorization rule

Element	Type	Optional	Description
ruleName	xsd:ID	No	A name associated with the rule. It is a key property of the rule and SHALL NOT be altered when included in the light-weight resource URL.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applToContacts	ApplicableToContacts	Yes	Defines for which contacts the rule applies for.
applToLists	ApplicableToLists	Yes	Defines for which lists the rule applies for.
applToListsAndContacts	ApplicableToListsAndLinkedContacts	Yes	Defines for which lists and referenced contacts the rule applies for.
authorizedTo	AuthorizedTo	No	Defines who have access to lists and contacts.
listFilter	xsd:string [0..unbounded]	Yes	Filter for list attributes (section 6.5) the retrieving user is allowed to see. An empty filter means that the user has access to all attributes.
indivFilter	xsd:string [0..unbounded]	Yes	Filter for member and contact attributes (section 6.1 and 6.9) the retrieving user is allowed to see. An empty filter means that the user has access to all attributes.
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named rule of type Rule is allowed in request and response bodies.

Regarding the clientCorrelator field, the note in section 5.2.2.12 applies.

5.2.2.16 Type: ApplicableToContacts

Contacts the rule applies to

Element	Type	Optional	Description
contactId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified list of 'contactId' that are part of the 'ContactCollection'.
allContacts	(empty)	Choice	The rule applies to all contacts.

XSD modelling uses an optional "choice" to select either contactId or allContacts.

5.2.2.17 Type: ApplicableToLists

Lists the rule applies to

Element	Type	Optional	Description
listId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified list of 'listId'.
allLists	(empty)	Choice	The rule applies to all lists.

XSD modelling uses an optional "choice" to select either listId or allLists.

5.2.2.18 Type: ApplicableToListsAndLinkedContacts

Lists and referenced contacts the rule applies to

Element	Type	Optional	Description
listId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified list of 'listId'. The rule applies also to contacts that are referenced from the specified list.
allLists	(empty)	Choice	The rule applies for all lists and referenced contacts.

XSD modelling uses an optional "choice" to select either listId or allLists.

5.2.2.19 Type: AuthorizedTo

Authorized identities

Element	Type	Optional	[ResourceRelPath]	Description
userId	xsd:anyURI [0..unbounded]	Choice	users/{userId}	User is authorized to access a resource for which the rule is applicable to.
allSharedIdentities	(empty)	Choice	*Not applicable*	All contacts with shared identities are authorized to access a resource for which the rule is applicable to.
listId	xsd:anyURI [0..unbounded]	Choice	lists/{listId}	Users within the list are authorized to access a resource for which the rule is applicable to.
allLists	(empty)	Choice	*Not applicable*	Users within all lists are authorized to access a resource for which the rule is applicable to.
link	common:Link [0..unbounded]	Choice	externalLists/{href}	Users within the list (owned by another user) are authorized to access a resource for which the rule is applicable to. Link specified using rel="List".

domainName	xsd:string [0..unbounded]	Choice	domains/{domainName}	Users within the domain are authorized to access a resource for which the rule is applicable to.
otherUser	(empty)	Choice	*Not applicable*	Users not authorized by any other rule are authorized to access a resource for which the rule is applicable to.

XSD modelling uses an optional "choice" to select either userId or allSharedIdentities or listId or allLists or link or domainName or otherUser.

5.2.2.20 Type: MemberTransferParameters

Set of parameters for the member transfer request

Element	Type	Optional	Description
destination	xsd:anyURI	No	Contains a relative URI representing the list where the member is to be transferred.

A root element named memberTransferParameters of type MemberTransferParameters is allowed in request bodies.

5.2.3 Enumerations

The subsections of this section define the enumerations used in the RESTful Address Book API.

5.2.3.1 Enumeration: ListType

Enumeration of types of lists

Enumeration	Description
URIList	Indicates that list is used as a list of URIs, such as a presence list, email distribution list.
GroupURIList	Indicates that list is used to store a list of URIs intended for group communication.
Group	Indicates that list is used for group communication, such as conferencing etc.

5.2.3.2 Enumeration: ResourceStatus

Enumeration of possible status of a subscription

Enumeration	Description
Active	Indicates that the subscription is active and authorized.
TerminatedTimeout	Indicates that the subscription has been terminated. The subscription was not refreshed in time before it expired.
Pending	Indicates that the subscription is awaiting an authorization decision.
TerminatedBlocked	Indicates that the subscription has been terminated. The subscription was blocked.
TerminatedNoResource	Indicates that the subscription has been terminated. The intended resource does not exist.
TerminatedOther	Indicates that the subscription has been terminated of an unknown reason.

5.2.4 Values of the Link "rel" attribute

The "rel" attribute of the Link element is a free string set by the server implementation, to indicate a relationship between the current resource and an external resource. The following are possible strings (list is non-exhaustive, and can be extended):

- Attribute
- AttributeList
- Contact
- AbChangesSubscription
- AbChangesSubscriptionCollection
- Member
- MemberCollection
- List
- ListCollection
- ListReferenceCollection
- MemberTransferParameters

These values indicate the kind of resource that the link points to.

5.3 Sequence Diagrams

The following sub-sections describe the resources, methods and steps involved in typical scenarios.

5.3.1 Managing contacts in the collection of contacts

The figure below shows various ways to retrieve and manage contacts in the collection of contacts. There is one application acting on behalf of userId. The application retrieves all contacts of userId, creates/updates a new contact for userId or deletes a contact for userId in the collection of contacts.

The resources:

- To retrieve all the user's contacts the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts

- To retrieve, add, update or delete one contact in the collection of contacts the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}

- To retrieve all attributes of a specific contact the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes

- To retrieve, create, update or delete a single attribute of a specific contact the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]

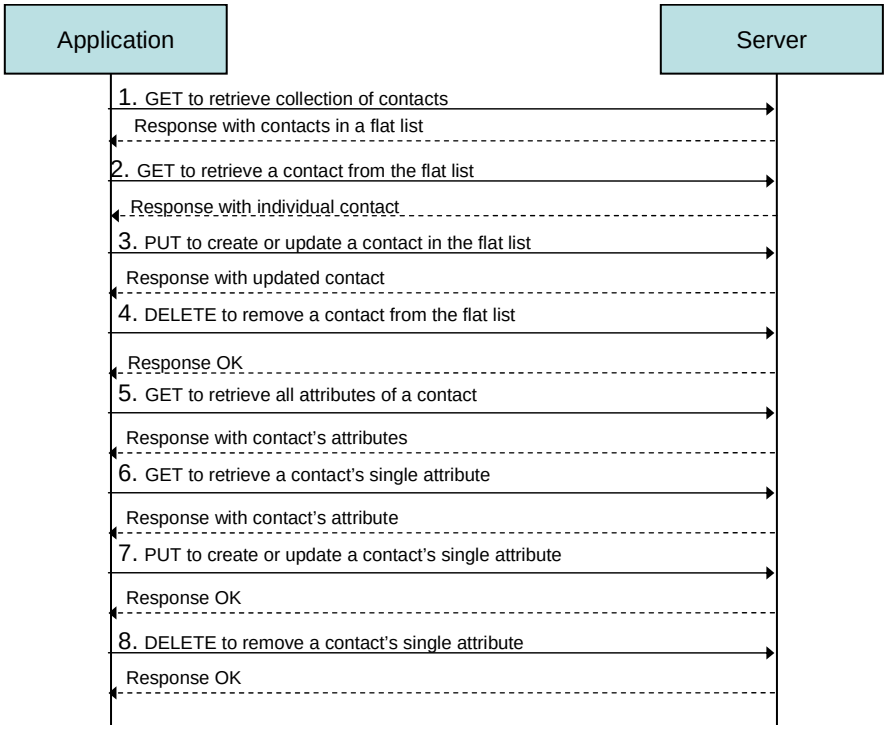

Figure 2 Managing contacts

1. The application retrieves all contacts for userId, by performing a GET on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts**

   The result contains all the contacts for userId.

2. The application retrieves one contact for userId, by performing a GET on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}**

   The result contains the information for one selected contact.

3. The application adds/updates a new contact by performing a PUT on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}**

   The result contains a return code indicating whether or not the operation was successful.

4. The application deletes one contact from the collection of all contacts by performing a DELETE on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}**

   The result contains a return code indicating whether or not the operation was successful.

5. The application retrieves all attributes of a specific contact by performing a GET on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes**

6. The application retrieves a single attribute of a specific contact by performing GET on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]**

7. The application creates or updates a single attribute of a specific contact by performing PUT on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]**

8. The application deletes a single attribute of a specific contact by performing DELETE on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]**

5.3.2 Accessing the lists and members

The figure below shows various ways to retrieve and manipulate data in lists. There is one application acting on behalf of Alice. The application is interested in one specific list, the list with id 'myFriends'. In that list there is one specific member, called Bob that is used in this example.

The resources:

- To fetch all the lists the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists

- To add a new list the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}

- To fetch all the member s of a list the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members

- To add a specific member to a list the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}

- To fetch all attributes for a list this resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/attributes

- To fetch all attributes of a specific member the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes

- To fetch a single attribute of a specific member the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]

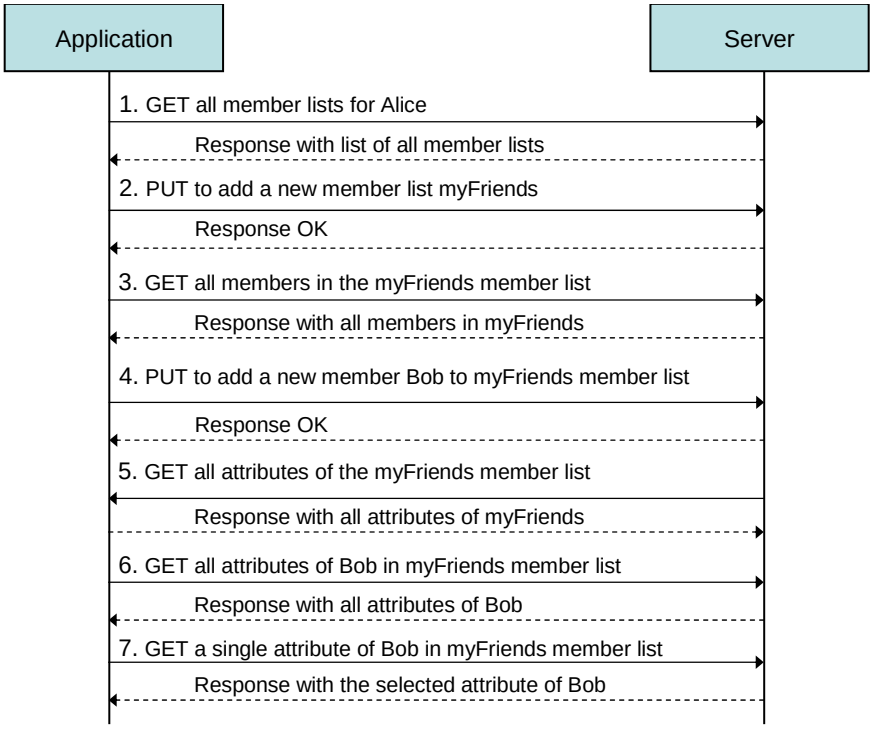

Figure 3 Accessing lists and members

.

1. The application fetches all the lists for the userId Alice by doing a GET on the following resource.
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists

   The result contains all the lists for Alice.

2. The application adds a new list called myFriends by doing a PUT on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends

   The result contains a return code indicating whether or not the operation was successful.

3. The application fetches all the members of the myFriends list by doing a GET on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/members

   The response will contain all the members of the myFriends list from Alice, but will not contain the attributes of the list.

4. The application adds a new member called Bob to the myFriends list doing a PUT on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/members/{bobUserId}

   The result contains a return code indicating whether or not the operation was successful.

5. The application fetches all attributes for the myFriends list by doing a GET on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/attributes

   The response will contain all attributes of the myFriends list.

6. The application fetches all attributes of Bob in the myFriends list by doing a GET on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/members/{bobUserId}/attributes

   The result contains all the attributes of Bob in this list.

7. The application fetches the display name attribute of Bob in the myFriends list by doing a GET on the following resource: **http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/members/{bobUserId}/attributes/display-name**

   The result contains Bob's display name attribute.

5.3.3 Subscribing to address book changes and receiving notifications

The figure below shows how a client can subscribe for notifications about contacts and list changes and receive notifications.

The notification URL passed by the client during the subscription step can be a Client-side Notification URL, or a Server-side Notification URL. Refer to [REST_NetAPI_NotificationChannel] for sequence flows illustrating the creation of a Notification Channel and obtaining a Server-side Notification URL on the server-side, and its use by the client via Long Polling.

The resources:

- To retrieve all the subscriptions to lists changes, and to create a new subscription the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/ subscriptions/abChanges

- To retrieve or delete an individual subscription, the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/subscriptions/abChanges/{subscriptionId}

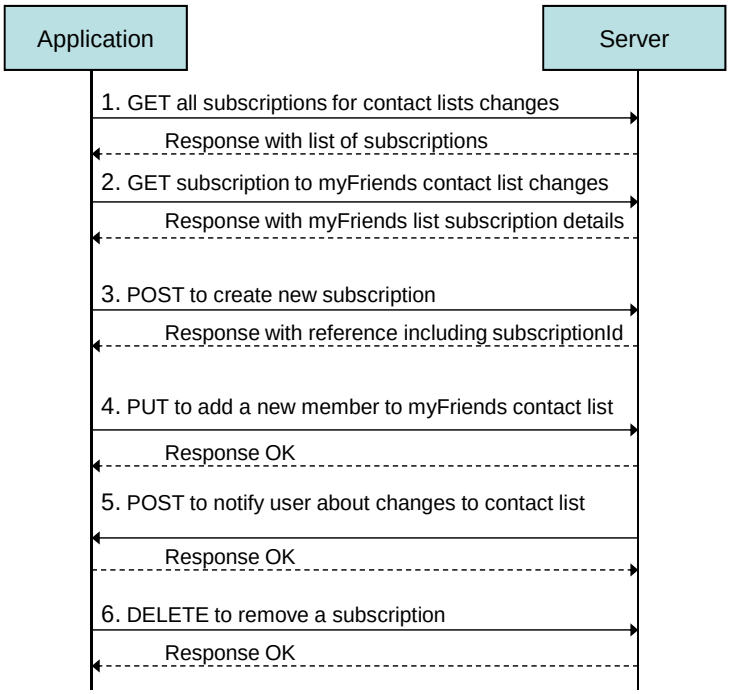

Figure 4 Subscription to notifications about address book changes

1. The application retrieves list of all subscriptions for list changes for the userId by performing a GET on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/subscriptions/abChanges

   The result contains all the subscriptions created by Alice.

2. The application retrieves an individual subscription identified by subscriptionId by performing a GET on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/subscriptions/abChanges/{subscriptionId}

   The result contains a return code indicating whether or not the operation was successful.

3. The application creates a new subscription for address book changes notification by performing a POST on the following resource, including in the request body a client application supplied notifyURL:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/subscriptions/abChanges

   If successful (as shown), the response will contain the complete representation of the newly created resource for subscription, including a self reference to the created resource.

4. The application adds a new member called Bob to the myFriends list doing a PUT on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId}/lists/myFriends/members/{bobUserId}

   The result contains a return code indicating whether or not the operation was successful.

5. The server notifies the application about the changes in his myFriends list, by performing a POST on the client application supplied notifyURL in the newly created subscriptionId:
   http://{client-supplied notifyURL}

   The result contains a return code indicating whether or not the operation was successful.

6. The application deletes the subscription identified by subscriptionId by performing a DELETE on the resource:
   http://{serverRoot}/addressbook/{apiVersion}/{aliceUserId} /subscription/abChanges/{subscriptionId}

   The result was successful.

5.3.4 Managing shared lists

The figure below shows how to manage shared lists.

The resources:

- To retrieve all lists shared by another user the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists

- To retrieve an individual list shared by another user the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}

- To retrieve a member's information from a list shared by another user the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}/members/{memberId}

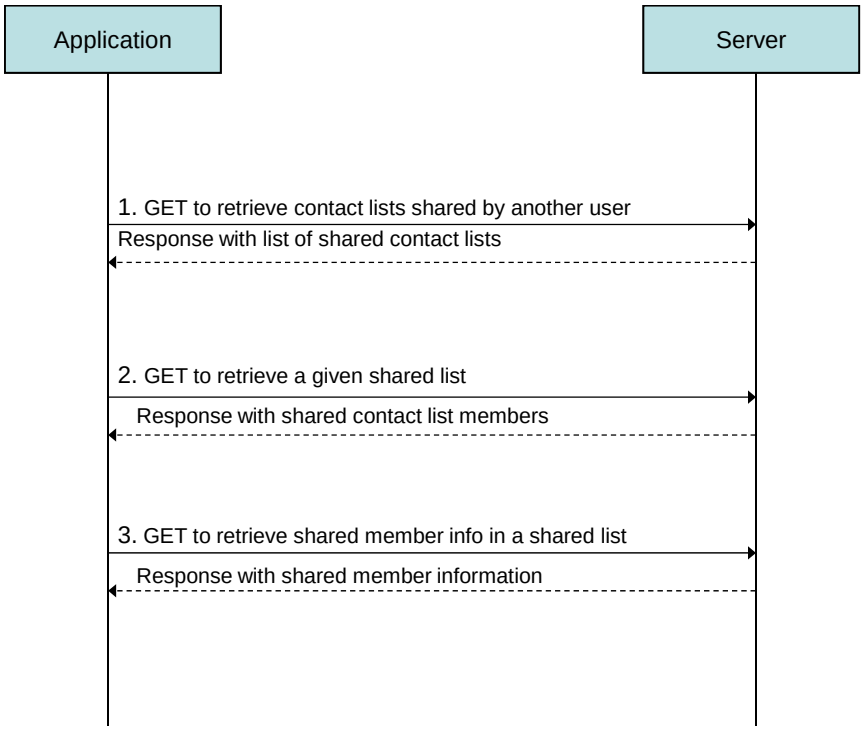

Figure 5 Flow for managing shared lists

1. The application retrieves a list of all lists shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists

   The result contains all the lists that otherUserId shares with userId.

2. The application retrieves an individual list shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}

   The result contains all the members in the shared list.

3. The application retrieves member's information from a list shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}/members/{sharedMemberId}

The result contains the selected member's information.

Note: sequence flows for managing shared contacts are similar.

5.3.5 Managing authorization rules

The figure below shows how to manage authorization rules for sharing lists.

The resources:

- To retrieve all authorization rules and to create a new authorization rule the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules

- To retrieve, update or delete an individual authorization rule the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}

- To retrieve, update or delete the authorized entity in an authorization rule the following resource is used:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]

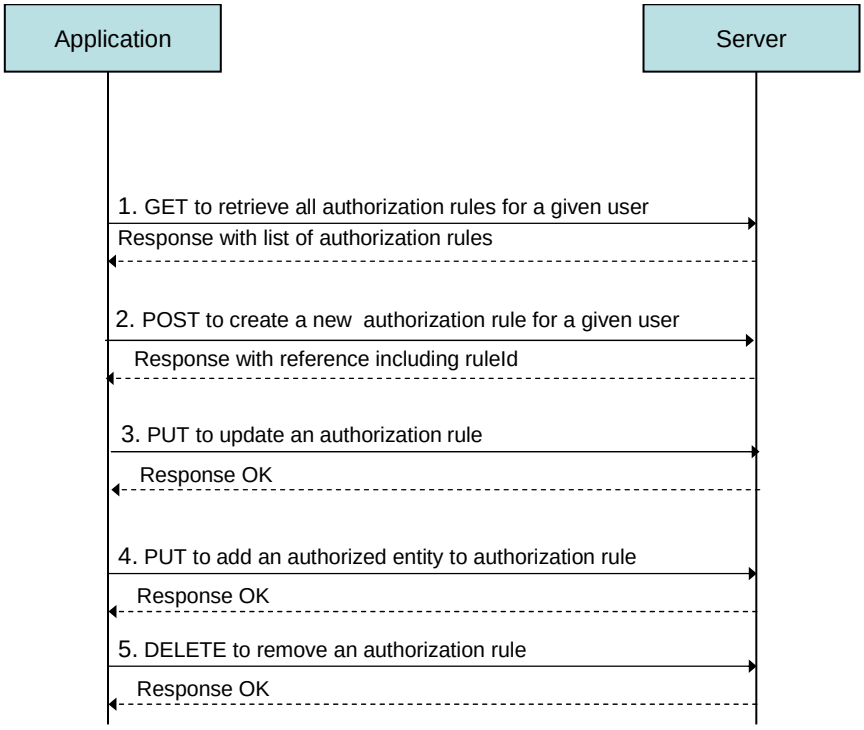

Figure 6 Flow for managing authorization rules

1. The application retrieves a list of all authroization rules for a given user lists shared by the user identified by userId, by performing a GET on the following resource:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules

   The result contains all the authorization rules for the given user.

2. The application creates a new authorization rule for a given user identified by userId, by performing a POST on the following resource:
http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules

   The result contains a reference to new resource including the newly created ruleId.

3. The application updates an authorization rule for the user identified by userId by performing a PUT on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}

   The result contains a return code indicating whether or not the operation was successful.

4. The application adds an authorized entity (e.g. Alice) to the authorization rule for the user identified by userId by performing a PUT on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}/users/{aliceUserId}

   The result contains a return code indicating whether or not the operation was successful.

5. The application deletes an authorization rule for the user identified by userId by performing a DELETE on the following resource:
   http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}

   The result contains a return code indicating whether or not the operation was successful.

6. Detailed specification of the resources

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML, JSON, application/x-www-form-urlencoded):

- 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 'Authorization' 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, and MAY support application/x-www-form-urlencoded 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: Collection of contacts

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts

This resource is used to retrieve a collection of all contacts owned by a given user.

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 Address Book API, see section 7.

6.1.3 GET

This resource is used to retrieve a collection of contacts and their information.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a contact in a collection of contacts, returned in a GET response body. If indivFilter is absent, GET response body SHALL include all contact sub-elements, plus all contact attributes (default behaviour). If indivFilterValue = Contact attribute name (e.g. name1), GET response body SHALL include all contact sub-elements, plus only selected contact attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all contact sub-elements and none of the contact attributes. If indivFilterValue = ~none, GET response body SHALL include none of the contact information (contact structure to be omitted). No other indivFilterValue is supported.

6.1.3.1 Example 1: Retrieve all contacts with all attributes (default) (Informative)

6.1.3.1.1 Request

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

6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contactCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contact>
    <contactId>alice</contactId>
    <attributeList>
    <attribute>
```

```
      <name>display-name</name>
      <value>Alice</value>
    </attribute>
    <attribute>
      <name>cellphone</name>
      <value>tel:+19585550109</value>
     </attribute>
     <attribute>
      <name>state</name>
      <value>California</value>
     </attribute>
     <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes</resourceURL>
   </attributeList>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice</resourceURL>
   <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122"/>
  </contact>
  <contact>
   <contactId>sam</contactId>
   <sharedIdentity>
     <sharedId>tel:+19585550121</sharedId>
   </sharedIdentity>
   <attributeList>
<attribute>
      <name>display-name</name>
      <value>Sam</value>
     </attribute>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550108</value>
     </attribute>
<attribute>
      <name>state</name>
      <value>New Jersey</value>
     </attribute>
     <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam/attributes</resourceURL>
   </attributeList>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam</resourceURL>
   <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550121"/>
   <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550121"/>
  </contact>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts</resourceURL>
</ab:contactCollection>
```

6.1.3.2 Example 2: Retrieve all contacts with selected attributes (Informative)

6.1.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

6.1.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contactCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <contact>
 <contactId>alice</contactId>
 <attributeList>
  <attribute>
     <name>cellphone</name>
     <value>tel:+19585550109</value>
   </attribute>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122"/>
 </contact>
 <contact>
  <contactId>sam</contactId>
  <sharedIdentity>
   <sharedId>tel:+19585550121</sharedId>
  </sharedIdentity>
  <attributeList>
<attribute>
     <name>cellphone</name>
     <value>tel:+19585550108</value>
   </attribute>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550121"/>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550121"/>
 </contact>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts</resourceURL>
</ab:contactCollection>
```

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 [RFC 2616].

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 [RFC 2616].

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 [RFC 2616].

6.2 Resource: Individual contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}

This resource is used to retrieve, create, update or delete individual contact from the collection of contacts of a given user.

6.2.1 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
contactId	contact identifier. Example: alice

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 Address Book API, see section 7.

6.2.3 GET

This resource is used to retrieve individual member data from the collection of contacts of a given user.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a contact in a collection of contacts, returned in a GET response body. If indivFilter is absent, GET response body SHALL include all contact sub-elements, plus all contact attributes (default behaviour). If indivFilterValue = Contact attribute name (e.g. name1), GET response body SHALL include all contact sub-elements, plus only selected contact attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. No other indivFilterValue is supported.

6.2.3.1 Example 1: Retrieve a contact with all attributes (default) (Informative)

6.2.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice HTTP/1.1
```

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

6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
<attribute>
       <name>display-name</name>
      <value>Alice</value>
</attribute><attribute>
      <name>cellphone</name>
      <value>tel:+19585550109</value>
    </attribute>
<attribute>
      <name>state</name>
      <value>California</value>
    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122"/>
</ab:contact>
```

6.2.3.2 Example 2: Retrieve a contact with selected attributes (Informative)

6.2.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
```

```
    <value>tel:+19585550109</value>
   </attribute>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122"/>
</ab:contact>
```

6.2.3.3 Example 3: Retrieve a contact with only vCard2.1 as selected attribute (Informative)

6.2.3.3.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice?indivFilter=vCard2.1 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.3.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
    <attribute>
      <name>vCard2.1</name>
      <objectValue>
QkVHSU46VkNBUkQNClZFUlNJT046Mi4xDQpOOkd1bXA7Rm9ycmVzdA0KRk46Rm9ycmVzdCBHdW1wDQpPUkc6QnViYmEgR3Vtc
CBTaHJpbXAgQ28uDQpUSVRMRTpTaHJpbXAgTWFuDQpURUw7V09SSztWT0lDRTooMTExKSA1NTUtMTIxMg0KVEVMO0hPTUU7
Vk9JQ0U6KDQwNCkgNTU1LTEyMTINCkFEUjtXT1JLOjs7MTAwIFdhdGVycyBFZGdlO0JheXRvd247TEE7MzAzMTQ7VW5pdGVkIFN
0YXRlcyBvZiBBbWVyaWNhDQpMQUJFTDtXT1JLO0VOQ09ESU5HPVFVT1RFRC1QUklOVEFCTEU6MTAwIFdhdGVycyBFZGdlPTBE
PTBBQmF5dG93biwgTEEgMzAzMTQ9MEQ9MEFVbml0ZWQgU3RhdGVzIG9mIEFtZXJpY2ENCkFEUjtIT01FOjs7NDIgUGxhbnRhdGlv
biBTdC47QmF5dG93bjtMQTszMDMxNDtVbml0ZWQgU3RhdGVzIG9mIEFtZXJpY2ENCkxBQkVMO0hPTUU7RU5DT0RJTkc9UVVPVE
VELVBSSU5UQUJMRTo0MiBQbGFudGF0aW9uIFN0Lj0wRD0wQUJheXRvd24sIExBIDMwMzE0PTBEPTBBVW5pdGVkIFN0YXRlcyBv
ZiBBbWVyaWNhDQpFTUFJTDtQUkVGO0lOVEVSTkVUOmZvcnJlc3RndW1wQGV4YW1wbGUuY29tDQpSRVY6MjAwODA0MjRUMTk
1MjQzWg0KRU5EOlZDQVJE
      </objectValue>
    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122"/>
</ab:contact>
```

6.2.4 PUT

This operation is used to create or update a contact in the collection of contacts.

6.2.4.1 Example 1: Create a new contact (Informative)

6.2.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>maria</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550106</value>
    </attribute>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria</resourceURL>
</ab:contact>
```

6.2.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>maria</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550106</value>
    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria</resourceURL>
</ab:contact>
```

6.2.4.2 Example 2: Update an existing contact with links to an existing member (Informative)

The PUT example is followed by a GET example on a list that was updated as a result of updating a Contact request. The Member is updated with the reverse link to the Contact.

6.2.4.2.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria HTTP/1.1
```

```
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>maria</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550106</value>
    </attribute>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550106"/>
</ab:contact>
```

6.2.4.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://example.com/v1/ addressbook/tel%3A%2B19585550100/contacts/maria
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>maria</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550106</value>
    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550106"/>
</ab:contact>
```

6.2.4.2.3 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.4.2.4 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
```

```
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>5678</listId>
  <memberCollection>
    <member>
      <memberId>+19585550106</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com
     </resourceURL>
     <link rel="Contact" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria"/>
   </member>
   <member>
     <memberId>mailto:liza@example.com</memberId>
     <attributeList>
       <attribute>
       <name>display-name</name>
         <value>Wife</value>
       </attribute>
       <attribute>
         <name>age</name>
         <value>42</value>
       </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com/attributes</resourceURL>
     </attributeList>
     <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com
     </resourceURL>
   </member>
   <member>
     <memberId>mailto:serezha@example.co</memberId>
     <attributeList>
       <attribute>
         <name>display-name</name>
         <value>Son</value>
       </attribute>
       <attribute>
         <name>phone</name>
         <value>tel:+19585550105</value>
       </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com/attributes</resourceURL>
     </attributeList>
     <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com
     </resourceURL>
   </member>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members</resourceURL>
 </memberCollection>
 <attributeList>
   <attribute>
     <name>display-name</name>
     <value>Family</value>
   </attribute>
```

```
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
</ab:list>
```

6.2.4.3 Example 3: Update an existing contact with links to non existing member (Informative)

In this example the user tries to update a contact including a link to a member which does not exists.

6.2.4.3.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>maria</contactId>
  <attributeList>
<attribute>
      <name>cellphone</name>
      <value>tel:+19585550106</value>
    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550108"/>
</ab:contact>
```

6.2.4.3.2 Response

```
HTTP/1.1 403 Forbidden
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

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, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.2.6 DELETE

This operation deletes a contact in a list. It SHALL also automatically delete any reverse links in a referenced member.

6.2.6.1 Example: Delete a contact (Informative)

6.2.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550105/contacts/sam HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.3 Resource: Attributes for a contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes

This resource is used to retrieve all attributes for a contact.

6.3.1 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
contactId	contact identifier. Example: alice

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

6.3.2 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 Address Book API, see section 7.

6.3.3 GET

This operation is used for retrieval of all attributes of a contact for a given user identity.

6.3.3.1 Example: Retrieve all attributes of a contact (Informative)

Retrieve all attributes of a contact belonging to a user, and return result in XML format, using resFormat parameter in URL.

6.3.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com
```

6.3.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attributeList xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
<attribute>
   <name>cellphone</name>
   <value>tel:+19585550106</value>
 </attribute>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes</resourceURL>
</ab:attributeList>
```

6.3.4 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 [RFC 2616].

6.3.5 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 [RFC 2616].

6.3.6 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 [RFC 2616].

6.4 Resource: Individual attribute for a contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]

This light-weight resource is used to manage attributes for a contact, which include creation, retrieval, update and delete operations.

6.4.1 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
contactId	contact identifier. Example: alice
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this light-weight resource is {name}, as defined in the [ResourceRelPath] column in table 5.2.2.6. {name} indicates the name of the attribute. Example: display-name

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

6.4.2 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 Address Book API, see section 7.

6.4.3 GET

This operation is used for retrieval of attribute values of a contact of a user.

6.4.3.1 Example 1: Retrieve a contact's attribute (Informative)

6.4.3.1.1 Request

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

6.4.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>cellphone</name>
  <value>tel:+19585550109</value>
</ab:attribute>
```

6.4.3.2 Example 2: Retrieve a contact's non existing attribute (Informative)

6.4.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attribute/alien  HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.3.2.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 12:55:59 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 Invalid input value for message part %1</text>
    <variables>Alien</variables>
  </serviceException>
</common:requestError>
```

6.4.4 PUT

This operation is used for creation or update of an attribute of a contact. If successful, this SHALL also create or update the attribute for the members in lists that are associated with the contact.

6.4.4.1 Example: Create or update a contact's attribute (Informative)

6.4.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes/married  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <name>married</name>
 <value>true</value>
</ab:attribute>
```

6.4.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes/married
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>married</name>
  <value>true</value>
</ab:attribute>
```

6.4.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: 'GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.4.6 DELETE

This operation deletes an attribute of a contact. If successful, this SHALL also delete the attribute from the members' attributes in lists associated with the contact.

6.4.6.1 Example: Delete a contact's attribute (Informative)

6.4.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes/display-name HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.5 Resource: Lists

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists

This resource is used to retrieve all lists belonging to a user.

6.5.1 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.

6.5.2 Response Codes

For HTTP response codes, see [REST_NetAPI_Common].

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

6.5.3 GET

This operation is used for retrieval of all lists for a given user identity.

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

Name	Type/Values	Optional	Description
listFilter	xsd:string	Yes	Defines the level of information related to a list, returned in a GET response body. If listFilter is absent, GET response body SHALL include all the list sub-elements, plus all list attributes (default behaviour). If listFilterValue = List attribute name (e.g. name1), GET response body SHALL include all list sub-elements, plus only selected list attributes (i.e. name1). Any number of listFilter=listFilterValue pairs separated by & are allowed. If listFilterValue = ~noAttr, GET response body SHALL include all the list sub-elements and none of the list attributes.

			No other listFilterValue is supported.
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.5.3.1 Example 1: Retrieve lists with all attributes and all members with all member attributes (default) (Informative)

6.5.3.1.1 Request

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

6.5.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <list>
  <listId>1234</listId>
  <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
    <attributeList>
     <attribute>
      <name>display-name</name>
      <value>Alice</value>
```

```
        </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes</resourceURL>
       </attributeList>       <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
     </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
   </memberCollection>
   <category>URIList</category>
   <sharedListIdentity>
     <sharedId>sip:bob@example.com;list=1234</sharedId>
   </sharedListIdentity>
   <attributeList>
     <attribute>
       <name>display-name</name>
       <value>Friends</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
   </attributeList>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
 </list>
 <list>
   <listId>5678</listId>
   <memberCollection>
     <member>
       <memberId>mailto:liza@example.com</memberId>
       <attributeList>
         <attribute>
           <name>display-name</name>
           <value>Wife</value>
         </attribute>
         <attribute>
           <name>age</name>
           <value>42</value>
         </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com/attributes</resourceURL>
       </attributeList>       <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com</resourceURL>
     </member>
     <member>
       <memberId>mailto:serezha@example.com</memberId>
       <attributeList>
         <attribute>
           <name>display-name</name>
           <value>Son</value>
         </attribute>
         <attribute>
           <name>phone</name>
           <value>tel:+19585550105</value>
         </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com/attributes</resourceURL>
       </attributeList>       <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com</resourceURL>
     </member>
```

```
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members</resourceURL>
   </memberCollection>
   <attributeList>
     <attribute>
       <name>display-name</name>
       <value>Family</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
   </attributeList>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
 </list>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.3.2 Example 2: Retrieve all lists belonging to a user without list attributes and without members (Informative)

Retrieve all lists belonging to a user, and return result in XML format.

6.5.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=~noAttr&indivFilter=~none HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <list>
   <listId>1234</listId>
   <category>URIList</category>
   <sharedListIdentity>
     <sharedId>sip:bob@example.com;list=1234</sharedId>
   </sharedListIdentity>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/mailto%3Abob@example.com/lists/1234</resourceURL>
 </list>
 <list>
   <listId>5678</listId>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/mailto%3Abob@example.com/lists/5678</resourceURL>
 </list>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.3.3 Example 3: Retrieve specific lists attributes and specific member attributes (Informative)

6.5.3.3.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=display-name&indivFilter=display-name&indivFilter=phone
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <list>
  <listId>1234</listId>
  <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
    <attributeList>
     <attribute>
      <name>display-name</name>
      <value>Alice</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes</resourceURL>
    </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
  </memberCollection>
  <category>URIList</category>
  <sharedListIdentity>
   <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Friends</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
 </list>
 <list>
  <listId>5678</listId>
  <memberCollection>
   <member>
    <memberId>mailto:liza@example.com</memberId>
```

```
      <attributeList>
       <attribute>
        <name>display-name</name>
        <value>Wife</value>
       </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com/attributes</resourceURL>
      </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com</resourceURL>
    </member>
    <member>
     <memberId>mailto:serezha@example.com</memberId>
     <attributeList>
      <attribute>
       <name>display-name</name>
       <value>Son</value>
      </attribute>
      <attribute>
       <name>phone</name>
       <value>tel:+19585550105</value>
      </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com/attributes</resourceURL>
      </attributeList>      <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com</resourceURL>
    </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members</resourceURL>
  </memberCollection>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Family</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
 </list>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.3.4 Example 4: Retrieve lists with specific lists attributes and without members data (Informative)

6.5.3.4.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=display-name&indivFilter=~none HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.4.2 Response

```
HTTP/1.1 200 OK
```

```
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <list>
    <listId>1234</listId>
    <category>URIList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com;list=1234</sharedId>
    </sharedListIdentity>
    <attributeList>
      <attribute>
        <name>display-name</name>
        <value>Friends</value>
      </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
  </list>
  <list>
    <listId>5678</listId>
    <attributeList>
      <attribute>
        <name>display-name</name>
        <value>Family</value>
      </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
  </list>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.3.5 Example 5: Retrieve specific lists attributes and list of members without attributes (Informative)

6.5.3.5.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=display-name&indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.5.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```
 <list>
  <listId>1234</listId>
  <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
  </memberCollection>
  <category>URIList</category>
  <sharedListIdentity>
   <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Friends</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
 </list>
 <list>
  <listId>5678</listId>
  <memberCollection>
   <member>
    <memberId>mailto:liza@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com</resourceURL>
   </member>
   <member>
    <memberId>mailto:serezha@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members</resourceURL>
  </memberCollection>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Family</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
 </list>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.3.6 Example 6: Retrieve lists without attributes and members with all member attributes (Informative)

6.5.3.6.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.3.6.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <list>
  <listId>1234</listId>
  <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
    <attributeList>
     <attribute>
      <name>display-name</name>
      <value>Alice</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aliza@example.com/attributes</resourceURL>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
 </list>
 <list>
  <listId>5678</listId>
  <memberCollection>
   <member>
    <memberId>mailto:liza@example.com</memberId>
    <attributeList>
     <attribute>
      <name>display-name</name>
      <value>Wife</value>
     </attribute>
     <attribute>
      <name>age</name>
      <value>42</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com/attributes</resourceURL>
```

```
      </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com</resourceURL>
    </member>
    <member>
      <memberId>mailto:serezha@example.com</memberId>
      <attributeList>
        <attribute>
          <name>display-name</name>
          <value>Son</value>
        </attribute>
        <attribute>
          <name>phone</name>
          <value>tel:+19585550105</value>
        </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com/attributes</resourceURL>
      </attributeList>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com</resourceURL>
    </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members</resourceURL>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678</resourceURL>
 </list>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists</resourceURL>
</ab:listCollection>
```

6.5.4 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 [RFC 2616].

6.5.5 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 [RFC 2616].

6.5.6 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 [RFC 2616].

6.6 Resource: Individual list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}

This resource is used to manage an individual list, which include creation, update, retrieval, and delete operations for a particular list.

6.6.1 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
listId	list identifier. Example: 1234

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

6.6.2 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 Address Book API, see section 7.

6.6.3 GET

This operation is used for retrieval of a given list. (This operation is normally only used to verify the existence of the list).

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

Name	Type/Values	Optional	Description
listFilter	xsd:string	Yes	Defines the level of information related to a list, returned in a GET response body. If listFilter is absent, GET response body SHALL include all the list sub-elements, plus all list attributes (default behaviour). If listFilterValue = List attribute name (e.g. name1), GET response body SHALL include all list sub-elements, plus only selected list attributes (i.e. name1). Any number of listFilter=listFilterValue pairs separated by & are allowed. If listFilterValue = ~noAttr, GET response body SHALL include all the list sub-elements and none of the list attributes. No other listFilterValue is supported.
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes.

			If indivFilterValue = ~none, GET response body SHALL include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.6.3.1 Example 1: Retrieve a list with all attributes and all member attributes (default) (Informative)

6.6.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <listId>1234</listId>
 <memberCollection>
   <member>
     <memberId>sip:alice@example.com</memberId>
     <attributeList>
       <attribute>
         <name>display-name</name>
         <value>Alice</value>
       </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes</resourceURL>
     </attributeList>
     <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
   </member>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
 </memberCollection>
 <attributeList>
   <attribute>
     <name>display-name</name>
     <value>Friends</value>
   </attribute>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
 </attributeList>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
</ab:list>
```

6.6.3.2 Example 2: Retrieve a list without list and member attributes (Informative)

6.6.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234?listFilter=~noAttr&indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <listId>1234</listId>
 <memberCollection>
    <member>
      <memberId>sip:alice@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
    </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
 </memberCollection>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
</ab:list>
```

6.6.3.3 Example 3: Retrieve a list with all attributes and selected member attributes (Informative)

6.6.3.3.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234?indivFilter=display-name HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.3.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <listId>1234</listId>
 <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
    <attributeList>
     <attribute>
```

```
        <name>display-name</name>
        <value>Alice</value>
      </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes</resourceURL>
    </attributeList>    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com</resourceURL>
  </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
 </memberCollection>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234</resourceURL>
</ab:list>
```

6.6.3.4 Example 4: Retrieve a non existing list (Informative)

6.6.3.4.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/4321 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.3.4.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 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">
 <link rel="List"
  href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/4321"/>
 <serviceException>
  <messageId>SVC0002</messageId>
  <text>Invalid input value for message part %1</text>
  <variables>listId</variables>
 </serviceException>
</common:requestError>
```

6.6.4 PUT

This operation is used for creation or update of a list.

6.6.4.1 Example: Create a list (Informative)

6.6.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
   <listId>bobPublic</listId>
   <memberCollection>
     <member>
        <memberId>mailto:alice@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members/
mailto%3Aalice@example.com</resourceURL>
     </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members</resourceURL>
   </memberCollection>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic</resourceURL>
</ab:list>
```

6.6.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
   <listId>bobPublic</listId>
   <memberCollection>
     <member>
        <memberId>mailto:alice@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members/
mailto%3Aalice@example.com</resourceURL>
     </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members</resourceURL>
   </memberCollection>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic</resourceURL>
</ab:list>
```

6.6.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: ’GET, PUT, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.6.6 DELETE

This operation deletes a list.

6.6.6.1 Example: Delete a list (Informative)

6.6.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic HTTP/1.1
Accept: application/xml
Host: example.com
```

6.6.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.7 Resource: Attributes for a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/attributes

This resource is used to retrieve all attributes of a list belonging to a user.

6.7.1 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.

6.7.2 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 Address Book API, see section 7.

6.7.3 GET

This operation is used for retrieval of all attributes for a list for a given user identity.

6.7.3.1 Example: Retrieve all attributes for a list belonging to Bob (Informative)

Retrieve all attributes for a list belonging to Bob, and return result in XML format.

6.7.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes HTTP/1.1
Accept: application/xml
Host: example.com
```

6.7.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attributeList xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <attribute>
```

```
   <name>display-name</name>
   <value>Friends</value>
  </attribute>
  <attribute>
   <name>date-created</name>
   <value>Fri, 10 Dec 2010 11:28:34 GMT</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes</resourceURL>
</ab:attributeList>
```

6.7.4 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 [RFC 2616].

6.7.5 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 [RFC 2616].

6.7.6 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 [RFC 2616].

6.8 Resource: Individual attribute for a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/attributes/[ResourceRelPath]

This resource is used to manage attributes on an individual list, which include creation, update, retrieval, and delete operations for the attributes.

6.8.1 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
listId	list identifier. Example: 4567
ResourceRelPath	Light-weight relative resource path. The allowed string for this light-weight resource is {name}, as defined in the [ResourceRelPath] column in table 5.2.2.6. {name} indicates the name of the attribute. Example: display-name

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

6.8.2 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 Address Book API, see section 7.

6.8.3 GET

This operation is used for retrieval of the value of a given attribute of a list.

6.8.3.1 Example 1: Retrieve an attribute value (Informative)

6.8.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/date-created HTTP/1.1
Accept: application/xml
Host: example.com
```

6.8.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>date-created</name>
  <value>Fri, 10 Dec 2010 11:28:34 GMT</value>
</ab:attribute>
```

6.8.3.2 Example 2: Retrieve a non existing attribute (Informative)

6.8.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/alien  HTTP/1.1
Accept: application/xml
Host: example.com
```

6.8.3.2.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 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>alien</variables>
  </serviceException>
</common:requestError>
```

6.8.4 PUT

This operation is used for creation or update of an attribute of a list.

6.8.4.1 Example: Create an attribute (Informative)

6.8.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/display-name  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>display-name</name>
  <value>Close friends</value>
</ab:attribute>
```

6.8.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://example.com/exampleAPI/v1/ addressbook/tel%3A%2B19585550123/lists/1234/attributes/display-name
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>display-name</name>
  <value>Close friends</value>
</ab:attribute>
```

6.8.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: ’GET, PUT, DELETE’ field in the response as per section 14.7 of [RFC 2616].

6.8.6 DELETE

This operation deletes an attribute of a list.

6.8.6.1 Example: Delete an attribute (Informative)

6.8.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/display-name  HTTP/1.1
Accept: application/xml
Host: example.com
```

6.8.6.1.2 Response

```
HTTP/1.1 204 No Content
```

Date: Fri, 10 Dec 2010 12:53:23 GMT

6.9 Resource: Members in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members

This resource is used to retrieve all members of a list belonging to a user.

6.9.1 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
listId	list identifier. Example: myList

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

6.9.2 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 Address Book API, see section 7.

6.9.3 GET

This operation is used for retrieval of all members of list for a given user identity.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL

			include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.9.3.1 Example 1: Retrieve all members for a list belonging to a user with all attributes (default) (Informative)

Retrieve all members of a list belonging to a user, and return result in XML format.

6.9.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members HTTP/1.1
Accept: application/xml
Host: example.com
```

6.9.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:memberCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <member>
  <memberId>mailto:alice@example.com</memberId>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Alice</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com
  </resourceURL>
 </member>
 <member>
  <memberId>tel:+19585550124</memberId>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Bob</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550124/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550124
  </resourceURL>
 </member>
```

```
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
</ab:memberCollection>
```

6.9.3.2 Example 2: Retrieve all members for a list belonging to a user, without member attributes (Informative)

Retrieve all members of a list belonging to a user, and return result in XML format.

6.9.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members?indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.9.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:memberCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <member>
    <memberId>mailto:alice@example.com</memberId>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com
    </resourceURL>
  </member>
  <member>
     <memberId>tel:+19585550124</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members
/tel%3A%2B19585550124</resourceURL>
  </member>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members</resourceURL>
</ab:memberCollection>
```

6.9.4 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 [RFC 2616].

6.9.5 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 [RFC 2616].

6.9.6 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 [RFC 2616].

6.10 Resource: Individual member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}

This resource is used to manage members on an individual list, which include creation, update, retrieval, and delete operations for the members.

6.10.1 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
listId	list identifier. Example: 6789
memberId	member identifier. Example: mailto:alice@example.com

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

6.10.2 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 Address Book API, see section 7.

6.10.3 GET

This operation is used for retrieval of a member from a list.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. No other indivFilterValue is supported.

6.10.3.1 Example 1: Retrieve a member (Informative)

6.10.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com HTTP/1.1
Accept: application/xml
Host: example.com
```

6.10.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
   <memberId>sip:alice@example.com</memberId>
  <attributeList>
    <attribute>
      <name>display-name</name>
      <value>Alice</value>
    </attribute>
    <attribute>
      <name>cellphone</name>
      <value>tel:+19585550155</value>
    </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com
  </resourceURL>
</ab:member>
```

6.10.3.2 Example 2: Retrieve a member and selected attributes (Informative)

6.10.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com?indivFilter=cellphone
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.10.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```
 <memberId>sip:alice@example.com</memberId>
 <attributeList>
  <attribute>
   <name>cellphone</name>
   <value>tel:+19585550155</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com/attributes</resourceURL>
 </attributeList>  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com</resourceURL>
</ab:member>
```

6.10.3.3 Example 3: Retrieve a non existing member (Informative)

6.10.3.3.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Ajoe@example.com  HTTP/1.1Accept:
application/xml
Host: example.com
```

6.10.3.3.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 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>memberId</variables>
 </serviceException>
</common:requestError>
```

6.10.4 PUT

This operation is used for creation or update of a member on a list.

6.10.4.1 Example 1: Create a member without a link to the contact (Informative)

6.10.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```
 <memberId>mailto:ole@example.com</memberId>
 <attributeList>
  <attribute>
   <name>display-name</name>
   <value>Ole</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
mailto%3Aole@example.com/attributes</resourceURL>
 </attributeList>
 <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
 mailto%3Aole@example.com</resourceURL>
</ab:member>
```

6.10.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
mailto%3A:ole@example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <memberId>mailto:ole@example.com</memberId>
 <attributeList>
  <attribute>
   <name>display-name</name>
   <value>Ole</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com/attributes</resourceURL>
 </attributeList>  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
mailto%3Aole@example.com
 </resourceURL>
</ab:member>
```

6.10.4.2 Example 2: Update member with relation to non-existing contact (Informative)

In this example the user tries to update a member including a link to a contact, which does not exists.

6.10.4.2.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Ajoe@example.com  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <memberId>mailto:joe@example.com</memberId>
 <attributeList>
  <attribute>
```

```
    <name>display-name</name>
    <value>Joe</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
mailto%3Ajoe@example.com/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
mailto%3Ajoe@example.com
  </resourceURL>
  <link rel="Contact" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/joe"/>
</ab:member>
```

6.10.4.2.2 Response

```
HTTP/1.1 403 Forbidden
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

6.10.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: 'GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.10.6 DELETE

This operation deletes a member from a list.

6.10.6.1 Example: Delete a member (Informative)

6.10.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/tel%3A%2B19585550125   HTTP/1.1
Accept: application/xml
Host: example.com
```

6.10.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.11 Resource: Attributes for a member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes

This resource is used to retrieve all attributes for a member of a list.

6.11.1 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
listId	list identifier. Example: myList.
memberId	member identifier. Example: tel:+19585550126

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

6.11.2 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 Address Book API, see section 7.

6.11.3 GET

This operation is used for retrieval of all members of list for a given user identity.

6.11.3.1 Example: Retrieve all attributes of a member of a list (Informative)

Retrieve all attributes of a member of a list belonging to a user, and return result in XML format.

6.11.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aalice@example.com/attributes HTTP/1.1
Accept: application/xml
Host: example.com
```

6.11.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attributeList xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <attribute>
    <name>display-name</name>
    <value>Alice</value>
  </attribute>
  <attribute>
    <name>cellphone</name>
    <value>tel:+19585550155</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members
/mailto%3Aalice@example.com/attributes</resourceURL>
</ab:attributeList>
```

6.11.4 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 [RFC 2616].

6.11.5 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 [RFC 2616].

6.11.6 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 [RFC 2616].

6.12 Resource: Individual attribute for a member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]

This light-weight resource is used to manage attributes for a member in a list, which include creation, update, retrieval, and delete operations.

6.12.1 Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. 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
listId	list identifier. Example: myFriends
memberId	member identifier. Example: tel:+19585550126
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this light-weight resource is {name}, as defined in the [ResourceRelPath] column in table 5.2.2.6. {name} indicates the name of the attribute. Example: display-name

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

6.12.2 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 Address Book API, see section 7.

6.12.3 GET

This operation is used for retrieval of attribute values of a memberon a list.

6.12.3.1 Example 1: Retrieve a member's attribute (Informative)

6.12.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/display-name
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.12.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>display-name</name>
  <value>Alice</value>
</ab:attribute>
```

6.12.3.2 Example 2: Retrieve a member's non existing attribute (Informative)

6.12.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/pet  HTTP/1.1
Accept: application/xml
Host: example.com
```

6.12.3.2.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 12:55:59 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 Invalid input value for message part %1</text>
    <variables>pet</variables>
  </serviceException>
</common:requestError>
```

6.12.4 PUT

This operation is used for creation or update of an attribute of a member on a list.

6.12.4.1 Example: Create a member's attribute (Informative)

6.12.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/born  HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <name>born</name>
```

```
  <value>1984</value>
</ab:attribute>
```

6.12.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/
sip%3Aalice@example.com/attributes/born
Content-Type: application/xml
Content-Length: nnnn


<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <name>born</name>
  <value>1984</value>
</ab:attribute>
```

6.12.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: 'GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.12.6 DELETE

This operation deletes a member from a list.

6.12.6.1 Example: Delete a member's attribute (Informative)

6.12.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/born
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.12.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.13 Resource: List references

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/listReferences

This resource is used to retrieve list of references for all nested lists.

6.13.1 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
listId	list identifier. Example: myList

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

6.13.2 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 Address Book API, see section 7.

6.13.3 GET

This operation is used for retrieval of a list of references to other lists.

6.13.3.1 Example: Retrieve all list references in a list (Informative)

Retrieve all list references in a list belonging to a user, and return result in XML format.

6.13.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences HTTP/1.1
Accept: application/xml
Host: example.com
```

6.13.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listReferenceCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/friends/listReferences</resourceURL>
  <link rel="List" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234"/>
  <link rel="List" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678"/>
</ab:listReferenceCollection>
```

6.13.4 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 [RFC 2616].

6.13.5 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 [RFC 2616].

6.13.6 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 [RFC 2616].

6.14 Resource: Individual list reference

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/listReferences/[ResourceRelPath]

This light-weight resource is used to manage list references in a nested list, which include creation, update, retrieval, and delete operations.

6.14.1 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
listId	list identifier. Example: myList
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this light-weight resource is {href}, as defined in the [ResourceRelPath] column in table 5.2.2.10. {href} indicates the link to the referenced list.

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

6.14.2 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 Address Book API, see section 7.

6.14.3 GET

This operation is used for retrieval of a list reference in a nested list. This is typically used to verify if the list reference exists.

6.14.3.1 Example 1: Retrieve a list reference (Informative)

6.14.3.1.1 Request

```
GET
/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/http%3A//example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.14.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:link xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1" rel="List"
```

```
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234"/>
```

6.14.3.2 Example 2: Retrieve a non existing list reference (Informative)

6.14.3.2.1 Request

```
GET
/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/http%3A//example.com/exampleAPI/addressbook/v
1/tel%3A%2B19585550100/lists/1235 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.14.3.2.2 Response

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 12:55:59 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 Invalid input value for message part %1</text>
    <variables>href</variables>
  </serviceException>
</common:requestError>
```

6.14.4 PUT

This operation is used for creation of a list reference in a list.

6.14.4.1 Example: Create a list reference (Informative)

6.14.4.1.1 Request

```
PUT
/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/http%3A//example.com/exampleAPI/addressbook/
v1/tel%3A%2B19585550100/lists/5678 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:link xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1" rel="List"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678"/>
```

6.14.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

```
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/memberlist/allFriends/listReferences/
http%3A//example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678
Content-Type: application/xml
Content-Length: nnnn


<?xml version="1.0" encoding="UTF-8"?>
<ab:link xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1" rel="List"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678"/>
```

6.14.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: 'GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.14.6 DELETE

This operation deletes a list reference in a list.

6.14.6.1 Example: Delete a list reference (Informative)

6.14.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/
        http%3A//example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.14.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.15 Resource: Address Book changes subscriptions

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/subscriptions/abChanges

This resource is used to retrieve address book changes subscriptions, as well as for the creation of new subscriptions.

This resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating a subscription.

6.15.1 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.

6.15.2 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 Address Book API, see section 7.

6.15.3 GET

This operation is used for retrieval of all list change subscriptions for a list.

6.15.3.1 Example: Retrieve all list changes subscriptions (Informative)

6.15.3.1.1 Request

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

6.15.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscriptionCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <abChangesSubscription>
    <listId>1234</listId>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <clientCorrelator>123</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>1958</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
  </abChangesSubscription>
  <abChangesSubscription>
    <anyContacts/>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/contacts</notifyURL>
      <callbackData>54321</callbackData>
    </callbackReference>
    <clientCorrelator>456</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>2641</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/123456789
</resourceURL>
  </abChangesSubscription>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges</resourceURL>
```

```
</ab:abChangesSubscriptionCollection>
```

6.15.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

6.15.5 POST

This operation is used for creation of a new subscription.

The notifyURL in the callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]).

6.15.5.1 Example 1: Create new subscription for list changes notification using 'tel' URI (Informative)

6.15.5.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
</ab:abChangesSubscription>
```

6.15.5.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
```

```
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

6.15.5.2 Example 2: Create new subscription for list changes notification using 'acr' URI (Informative)

6.15.5.2.1 Request

```
POST /exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
</ab:abChangesSubscription>
```

6.15.5.2.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

6.15.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

6.16 Resource: Individual list changes subscription

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/subscriptions/abChanges/{subscriptionId}

This resource is used to retrieve or remove an individual list changes subscription.

6.16.1 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
subscriptionId	identifier associated with this specific subscription

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

6.16.2 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 Address Book API, see section 7.

6.16.3 GET

This operation is used for retrieval of a specific list changes subscription.

6.16.3.1 Example: Retrieve individual list changes subscription (Informative)

Retrieve a specific list changes subscription.

6.16.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.16.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010  21:34:52 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
```

```
  <applicationTag>myApp</applicationTag>
  <duration>6817</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

6.16.4 PUT

This operation is used to update and/or to extend the duration of a subscription.

6.16.4.1 Example: Update/Extend duration of subscription (Informative)

6.16.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321 HTTP/1.1
Host: example.com
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

6.16.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

6.16.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.16.6 DELETE

This operation deletes a subscription to list changes.

6.16.6.1 Example: Delete a list changes subscription (Informative)

6.16.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.16.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.17 Resource: Client resource for list changes notifications

This resource is a callback URL provided by the client for notification about outbound message delivery status. The RESTful Address Book API does not make any assumption about the structure of this URL. If this URL is a Client-side Notification URL, the server will POST notifications directly to it. If this URL is a Server-side Notification URL, the server uses it to determine the address of the Notification Server to which the notifications will subsequently be POSTed. The way the server determines the address of the Notification Server is out of scope of this specification.

Note: In the case when the client has set up a Notification Channel in order to use Long Polling to obtain the notifications, in order to retrieve the notifications, the client needs to use the Long Polling mechanism described in [REST_NetAPI_NotificationChannel], instead of the mechanism described below in section 6.17.5.

A notification SHALL be generated by the system in the following occasions:

Type of notification	Value of ResourceStatus	Generated in the following occasions:
Update number of contacts or members	Active	A contact is added/removed to/from the contact collection or a member is added/removed to/from a list
Remove contacts or members	Active	A contact is removed from the contact collection or a member is removed from a list
Update contact or member attributes	Active	A contact's or a member's attributes are updated (attribute added, changed or removed)
Update contact collection or list attributes	Active	A contact collection's or a list's attributes are updated (attribute added, changed or removed)
Final notification	TerminatedTimeout	The subscription has expired (time indicated by 'duration' has elapsed). Note: A request to delete the subscription does not generate a new notification.
	TerminatedNoResource	The subscription has been terminated since the resource

		(e.g. specified list) has been removed.
	TerminatedOther	The subscription has been terminated for an unknown reason.

- to indicate that the subscription has been terminated by the server.
(ResourceStatus=TerminatedTimeout/TerminatedNoResource/TerminatedOther)

6.17.1 Request URL variables

Client provided.

6.17.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

6.17.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

6.17.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

6.17.5 POST

This operation is used to send notification about list change to the application.

6.17.5.1 Example 1: Notification for subscription with sendFullListContent=true (default) (Informative)

6.17.5.1.1 Request

```
POST /notifications/lists/1234 HTTP/1.1
Accept: application/xml
Host: application.example.com
Content-Type: application/xml
Content-Length:nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesNotification xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <callbackData>12345</callbackData>
  <resourceStatus>Active</resourceStatus>
   <duration>3480</duration>
   <link rel="Member" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/
     members/sip%3Aalice@example.com"/>
   <link rel="AbChangesSubscription"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"/>
</ab:abChangesNotification>
```

6.17.5.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

6.17.5.2 Example 2: Notification for contact change subscription (Informative)

6.17.5.2.1 Request

```
POST /notifications/contacts HTTP/1.1
Accept: application/xml
Host: application.example.com
Content-Type: application/xml
Content-Length: nnnn


<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesNotification xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <callbackData>54321</callbackData>
  <resourceStatus>Active</resourceStatus>
  <duration>4217</duration>
  <link rel="Contact" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/ole"/>
  <link rel="AbChangesSubscription"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/123456789"/>
</ab:abChangesNotification>
```

6.17.5.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

6.17.5.3 Example 3: Notification for expired contact subscription (Informative)

6.17.5.3.1 Request

```
POST /notifications/contacts HTTP/1.1
Accept: application/xml
Host: application.example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesNotification xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <callbackData>54321</callbackData>
  <resourceStatus>TerminatedTimeout</resourceStatus>
   <link rel="AbChangesSubscription" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/
abChanges/123456789"/>
</ab:abChangesNotification>
```

6.17.5.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

6.17.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

6.18 Resource: Collection of contacts shared by other user

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/contacts

This resource is used to retrieve contacts shared by user identified by otherUserId.

6.18.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	sharing user identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.18.2 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 Address Book API, see section 7.

6.18.3 GET

This operation is used for retrieval of all contacts shared by user identified by otherUserId.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a contact in a contact collection, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all contact sub-elements, plus all contact attributes (default behaviour). If indivFilterValue = Contact attribute name (e.g. name1), GET response body SHALL include all contact sub-elements, plus only selected contact attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all contact sub-elements and none of the contact attributes.

			If indivFilterValue = ~none, GET response body SHALL include none of the contact information (contact structure to be omitted). No other indivFilterValue is supported.

6.18.3.1 Example 1: Retrieve contacts shared by a user identified by otherUserId with all contact attributes (default) (Informative)

6.18.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts HTTP/1.1
Accept: application/xml
Host: example.com
```

6.18.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contactCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <contact>
  <contactId>alice</contactId>
  <sharedIdentity>
   <sharedId>sip:alice@example.com</sharedId>
  </sharedIdentity>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Alice</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
sip%3Aalice@example.com/attributes
</resourceURL>
  </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts
/sip%3Aalice@example.com</resourceURL>
 </contact>
 <contact>
  <contactId>ole</contactId>
  <sharedIdentity>
   <sharedId>mailto:ole@example.com</sharedId>
  </sharedIdentity>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Ole</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
mailto%3Aole@example.com/
attributes</resourceURL>
```

```
   </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
mailto%3Aole@example.com
</resourceURL>
   <link rel="Member" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/shared/lists/6789/members/
mailto%3Aole@example.com"/>
  </contact>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts
</resourceURL>
</ab:contactCollection>
```

6.18.3.2 Example 2: Retrieve contacts shared by a user identified by otherUserId without contacts attributes

(Informative)

6.18.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts}?indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.18.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contactCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contact>
    <contactId>alice</contactId>
    <sharedIdentity>
      <sharedId>sip:alice@example.com</sharedId>
    </sharedIdentity>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
sip%3Aalice@example.com</resourceURL>
  </contact>
  <contact>
    <contactId>ole</contactId>
    <sharedIdentity>
      <sharedId>mailto:ole@example.com</sharedId>
    </sharedIdentity>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
mailto%3Aole@example.com</resourceURL>
    <link rel="Member" href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/shared/lists/6789/members/
mailto%3Aole@example.com"/>
  </contact>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts
</resourceURL>
</ab:contactCollection>
```

6.18.4 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 [RFC 2616].

6.18.5 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 [RFC 2616].

6.18.6 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 [RFC 2616].

6.19 Resource: Individual shared contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/contacts/{sharedIdentity}

This resource is used to retrieve a contact shared by user identified by otherUserId.

6.19.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	sharing user identifier. Examples: tel:+19585550100, acr:pseudonym123
sharedIdentity	identifier of the shared contact. Examples: alice, mailto:alice@example.com

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

6.19.2 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 Address Book API, see section 7.

6.19.3 GET

This resource is used to retrieve individual data about the shared contact.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a contact in a contact collection, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all contact sub-elements, plus all contact attributes (default behaviour).

			If indivFilterValue = Contact attribute name (e.g. name1), GET response body SHALL include all contact sub-elements, plus only selected contact attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all contact sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL include none of the contact information (contact structure to be omitted). No other indivFilterValue is supported.

6.19.3.1 Example 1: Retrieve information about individual shared contact using contactId (Informative)

In this example the contactId is known to the requestor.

6.19.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B 19585550105/shared/tel%3A%2B19585550100/contacts/alice HTTP/1.1
Accept: application/xml
Host: example.com
```

6.19.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
    <attribute>
      <name>display-name</name>
      <value>Alice</value>
    </attribute>
    <attribute>
      <name>cellphone</name>
      <value>tel:+19585550109</value>
    </attribute>
    <attribute>
      <name>state</name>
      <value>California</value>
    </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
alice/attributes</resourceURL>
  </attributeList>
```

```
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/alice
</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/
1234/members/tel%3A%2B19585550122"/>
</ab:contact>
```

6.19.3.2 Example 2: Retrieve information about individual shared contact using sharedIdentity (Informative)

In this example the requestor uses the publicly known address (sharedIdentity) of the shared contact to be retrieved.

6.19.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/mailto%3Aalice@example.com
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.19.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
<attribute>
     <name>display-name</name>
     <value>Alice</value>
   </attribute>
<attribute>
     <name>cellphone</name>
     <value>tel:+19585550109</value>
   </attribute>
   <attribute>
     <name>state</name>
     <value>California</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
alice/attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/alice
</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550122"/>
</ab:contact>
```

6.19.3.3 Example 3: Retrieve information about individual shared contact, with selected contact attributes (default) (Informative)

6.19.3.3.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B 19585550105/shared/tel%3A%2B19585550100/contacts/alice?indivFilter=cellphone
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.19.3.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:contact xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <contactId>alice</contactId>
  <attributeList>
<attribute>
     <name>cellphone</name>
     <value>tel:+19585550109</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
alice/attributes</resourceURL>
  </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/alice
</resourceURL>
  <link rel="Member"
href="http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/tel
%3A%2B19585550122"/>
  </ab:contact>
```

6.19.4 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 [RFC 2616].

6.19.5 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 [RFC 2616].

6.19.6 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 [RFC 2616].

6.20 Resource: Collection of shared lists

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists

This resource is used to retrieve lists shared by user identified by otherUserId.

6.20.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	sharing user identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.20.2 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 Address Book API, see section 7.

6.20.3 GET

This operation is used for retrieval of all lists shared by user identified by otherUserId.

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

Name	Type/Values	Optional	Description
listFilter	xsd:string	Yes	Defines the level of information related to a list, returned in a GET response body. If listFilter is absent, GET response body SHALL include all the list sub-elements, plus all list attributes (default behaviour). If listFilterValue = List attribute name (e.g. name1), GET response body SHALL include all list sub-elements, plus only selected list attributes (i.e. name1). Any number of listFilter=listFilterValue pairs separated by & are allowed. If listFilterValue = ~noAttr, GET response body SHALL include all the list sub-elements and none of the list attributes. No other listFilterValue is supported.
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET

			response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.20.3.1 Example 1: Retrieve lists shared by a user identified by otherUserId with all list attributes and all member attributes (default) (Informative)

6.20.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists HTTP/1.1
Accept: application/xml
Host: example.com
```

6.20.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <list>
  <listId>1234</listId>
  <memberCollection>
   <member>
    <memberId>sip:alice@example.com</memberId>
    <attributeList>
     <attribute>
      <name>display-name</name>
      <value>Alice</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com/
attributes</resourceURL>
    </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com
</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members</resourceURL>
```

```
    </memberCollection>
    <category>URIList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com;list=1234</sharedId>
    </sharedListIdentity>
    <attributeList>
      <attribute>
        <name>display-name</name>
        <value>Friends</value>
      </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/
attributes</resourceURL>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234
</resourceURL>
  </list>
  <list>
    <listId>5678</listId>
    <memberCollection>
      <member>
        <memberId>mailto:liza@example.com</memberId>
        <attributeList>
          <attribute>
            <name>display-name</name>
            <value>Wife</value>
          </attribute>
          <attribute>
            <name>age</name>
            <value>42</value>
          </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aliza@example.com/
attributes</resourceURL>
        </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aliza@example.com
</resourceURL>
      </member>
      <member>
        <memberId>mailto:serezha@example.com</memberId>
        <attributeList>
          <attribute>
            <name>display-name</name>
            <value>Son</value>
          </attribute>
          <attribute>
            <name>phone</name>
            <value>tel:+19585550105</value>
          </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aserezha@example.com/attributes</resourceURL>
        </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aserezha@example.com
</resourceURL>
      </member>
```

```
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/members</resourceURL>
   </memberCollection>
   <attributeList>
     <attribute>
       <name>display-name</name>
       <value>Family</value>
     </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/attributes</resourceURL>
   </attributeList>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678</resourceURL>
  </list>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists
  </resourceURL>
</ab:listCollection>
```

6.20.3.2 Example 2: Retrieve lists shared by a user identified by otherUserId without lists attributes, but with selected member attributes (Informative)

6.20.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists?listFilter=~noAttr&indivFilter=phone HTTP/1.1
Accept: application/xml
Host: example.com
```

6.20.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <list>
    <listId>1234</listId>
    <memberCollection>
      <member>
        <memberId>sip:alice@example.com</memberId>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com
        </resourceURL>
      </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members</resourceURL>
    </memberCollection>
    <category>URIList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com;list=1234</sharedId>
```

```
   </sharedListIdentity>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234
</resourceURL>
  </list>
  <list>
   <listId>5678</listId>
   <memberCollection>
    <member>
      <memberId>mailto:liza@example.com</memberId>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aliza@example.com
</resourceURL>
    </member>
    <member>
     <memberId>mailto:serezha@example.com</memberId>
     <attributeList>
      <attribute>
       <name>phone</name>
       <value>tel:+19585550105</value>
      </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aserezha@example.com/
attributes</resourceURL>
     </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers/mailto%3Aserezha@example.com
</resourceURL>
    </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/me
mbers</resourceURL>
   </memberCollection>
   <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678
</resourceURL>
  </list>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists
  </resourceURL>
</ab:listCollection>
```

6.20.4 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 [RFC 2616].

6.20.5 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 [RFC 2616].

6.20.6 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 [RFC 2616].

6.21 Resource: Individual shared list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}

This resource is used to retrieve individual list identified by sharedListIdentity shared by user identified by otherUserId with user identified by userId.

6.21.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	sharing user identifier. Examples: tel:+19585550100, acr:pseudonym123
sharedListIdentity	shared list identifier. Example: myList

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

6.21.2 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 Address Book API, see section 7.

6.21.3 GET

This operation is used for retrieval of all individual shared list information for a given user identity.

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

Name	Type/Values	Optional	Description
listFilter	xsd:string	Yes	Defines the level of information related to a list, returned in a GET response body. If listFilter is absent, GET response body SHALL include all the list sub-elements, plus all list attributes (default behaviour). If listFilterValue = List attribute name (e.g. name1), GET response body SHALL include all list sub-elements, plus only selected list attributes (i.e. name1). Any number of listFilter=listFilterValue pairs separated by & are allowed. If listFilterValue = ~noAttr, GET response body SHALL include all the list sub-elements and none of the list attributes. No other listFilterValue is supported.
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL

			include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.21.3.1 Example 1: Retrieve individual list shared by a user identified by otherUserId, with all list attributes and all member attributes (default) (Informative)

6.21.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.21.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <listId>1234</listId>
 <memberCollection>
  <member>
   <memberId>sip:alice@example.com</memberId>
   <attributeList>
    <attribute>
     <name>display-name</name>
     <value>Alice</value>
    </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com/
attributes</resourceURL>
   </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/me
```

```
mbers/sip%3Aalice@example.com
</resourceURL>
   </member>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/me
mbers</resourceURL>
  </memberCollection>
  <category>URIList</category>
  <sharedListIdentity>
    <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
    <attribute>
      <name>display-name</name>
      <value>Friends</value>
    </attribute>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/
attributes</resourceURL>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234
</resourceURL>
</ab:list>
```

6.21.3.2 Example 2: Retrieve individual list shared by a user identified by otherUserId, with all list attributes and no member attributes (default) (Informative)

6.21.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234?indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com
```

6.21.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:list xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <memberCollection>
    <member>
      <memberId>sip:alice@example.com</memberId>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/
        members/sip%3Aalice@example.com</resourceURL>
    </member>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/
        members</resourceURL>
  </memberCollection>
```

```
 <category>URIList</category>
 <sharedListIdentity>
  <sharedId>sip:bob@example.com;list=1234</sharedId>
 </sharedListIdentity>
 <attributeList>
  <attribute>
   <name>display-name</name>
   <value>Friends</value>
  </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/
attributes</resourceURL>
 </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234
</resourceURL>
</ab:list>
```

6.21.4 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 [RFC 2616].

6.21.5 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 [RFC 2616].

6.21.6 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 [RFC 2616].

6.22 Resource: Members in a shared list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}/members

This resource is used to retrieve all members in the list identified by sharedListIdentity shared by user identified by otherUserId with user identified by userId.

6.22.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	sharing user identifier. Examples: tel:+19585550100, acr:pseudonym123
sharedListIdentity	shared list identifier. Example: myList

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

6.22.2 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 Address Book API, see section 7.

6.22.3 GET

This operation is used for retrieval of all members of a shared list for a given user identity.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. If indivFilterValue = ~noAttr, GET response body SHALL include all member sub-elements and none of the member attributes. If indivFilterValue = ~none, GET response body SHALL include none of the member information (member structure to be omitted). No other indivFilterValue is supported.

6.22.3.1 Example: Retrieve all members for a shared list belonging to a user with all attributes (default) (Informative)

6.22.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members HTTP/1.1
Accept: application/xml
Host: example.com
```

6.22.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
```

```
<ab:memberCollection xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <member>
  <memberId>mailto:alice@example.com</memberId>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Alice</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com/attributes</resourceURL>
  </attributeList>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
mailto%3Aalice@example.com
  </resourceURL>
 </member>
 <member>
  <memberId>tel:+19585550124</memberId>
  <attributeList>
   <attribute>
    <name>display-name</name>
    <value>Bob</value>
   </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550124/attributes</resourceURL>
  </attributeList>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550124
  </resourceURL>
 </member>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members</resourceURL>
</ab:memberCollection>
```

6.22.4 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 [RFC 2616].

6.22.5 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 [RFC 2616].

6.22.6 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 [RFC 2616].

6.23 Resource: Individual member information from shared list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/sharedBy/{otherUserId}/{sharedListId}/members/{sharedMemberId}

This resource is used to retrieve individual member information from the list shared by user identified by otherUserId.

6.23.1 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	requesting user identifier. Examples: tel:+19585550105, acr:pseudonym456
otherUserId	other user identifier. Examples: tel:+19585550100, acr:pseudonym123
sharedListIdentity	shared list identifier. Examples: myList
memberId	list member identifier. Example: mailto:alice@example.com

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

6.23.2 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 Address Book API, see section 7.

6.23.3 GET

This operation is used for retrieval of all information about a member in a shared list.

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

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	Defines the level of information related to a member in a list, returned in a GET response body. If indivFilterValue is absent, GET response body SHALL include all member sub-elements, plus all member attributes (default behaviour). If indivFilterValue = Member attribute name (e.g. name1), GET response body SHALL include all member sub-elements, plus only selected member attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed. No other indivFilterValue is supported.

6.23.3.1 Example 1: Retrieve all member attributes about user identified as memberId from the list shared by a user identified by otherUserId (default)(Informative)

6.23.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com HTTP/1.1
Accept: application/xml
Host: example.com
```

6.23.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <memberId>sip:alice@example.com</memberId>
  <attributeList>
    <attribute>
      <name>display-name</name>
      <value>Alice</value>
    </attribute>
    <attribute>
      <name>cellphone</name>
      <value>19585550105</value>
    </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/me
mbers/sip%3Aalice@example.com
/attributes</resourceURL>
  </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/me
mbers/sip%3Aalice@example.com
  </resourceURL>
</ab:member>
```

6.23.3.2 Example 2: Retrieve selected member attributes about user identified as memberId from the list shared by a user identified by otherUserId (default) (Informative)

6.23.3.2.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

6.23.3.2.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

```
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:member xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <memberId>sip:alice@example.com</memberId>
  <attributeList>
    <attribute>
      <name>cellphone</name>
      <value>tel:+19585550105</value>
    </attribute>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com/
attributes</resourceURL>
  </attributeList>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com
  </resourceURL>
</ab:member>
```

6.23.4 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 [RFC 2616].

6.23.5 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 [RFC 2616].

6.23.6 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 [RFC 2616].

6.24 Resource: Authorization rules

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules

This resource is used to retrieve rules for access to shared lists, as well as for the creation of new authorization rules.

6.24.1 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.

6.24.2 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 Address Book API, see section 7.

6.24.3 GET

This operation is used for retrieval of all authorization rules.

6.24.3.1 Example: Retrieve all authorization rules (Informative)

6.24.3.1.1 Request

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

6.24.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:ruleList xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
 <rule>
    <ruleName>allowAllContacts</ruleName>
    <clientCorrelator>123</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <applToContacts>
      <allContacts/>
    </applToContacts>
    <authorizedTo>
      <listId>1234</listId>
    </authorizedTo>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1</resourceURL>
 </rule>
 <rule>
    <ruleName>allowOwnContact</ruleName>
    <clientCorrelator>456</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <applToContacts>
      <contactId>bob</contactId>
    </applToContacts>
    <authorizedTo>
      <allSharedIdentities/>
    </authorizedTo>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2</resourceURL>
 </rule>
 <rule>
    <ruleName>allowFriendsList</ruleName>
    <clientCorrelator>789</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <applToLists>
      <listId>1234</listId>
```

```
    </applToLists>
    <authorizedTo>
      <listId>1234</listId>
    </authorizedTo>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3</resourceURL>
  </rule>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules</resourceURL>
</ab:ruleList>
```

6.24.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

6.24.5 POST

This operation is used for creation of a new authorization rule.

6.24.5.1 Example: Create new rule for list and referenced contacts (Informative)

This rule allows users in the specified list (authorizedTo element) to retrieve the members in the same list and also the contacts that have references to the members in the list.

6.24.5.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <ruleName>combinedRule</ruleName>
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToListsAndContacts>
    <listId>5678</listId>
  </applToListsAndContacts>
  <authorizedTo>
    <listId>5678</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3</resourceURL>
</ab:rule>
```

6.24.5.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```
  <ruleName>combinedRule</ruleName>
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToListsAndContacts>
    <listId>5678</listId>
  </applToListsAndContacts>
  <authorizedTo>
    <listId>5678</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3</resourceURL>
</ab:rule>
```

6.24.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

6.25 Resource: Individual authorization rule

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}

This resource is used to retrieve an individual authorization rule for access to shared lists.

6.25.1 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
ruleId	rule identifier: Example: 987654321

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

6.25.2 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 Address Book API, see section 7.

6.25.3 GET

This operation is used for retrieval of an individual authorization rule.

6.25.3.1 Example: Retrieve an authorization rule (Informative)

6.25.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.25.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <ruleName>allowAllContacts</ruleName>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
  <authorizedTo>
    <listId>1234</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1</resourceURL>
</ab:rule>
```

6.25.4 PUT

This operation is used to update an individual authorization rule.

6.25.4.1 Example: Update an existing authorization rule (Informative)

6.25.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <ruleName>allowAllContacts</ruleName>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
  <authorizedTo>
    <listId>1234</listId>
    <listId>5678</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1</resourceURL>
</ab:rule>
```

6.25.4.1.2 Response

```
HTTP/1.1 200 OK
```

```
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length : nnnn


<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <ruleName>allowAllContacts</ruleName>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
  <authorizedTo>
    <listId>1234</listId>
    <listId>5678</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1</resourceURL>
</ab:rule>
```

6.25.5 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 [RFC 2616].

6.25.6 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 [RFC 2616].

6.25.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.25.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.26 Resource: Individual authorization rule data

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]

This resource is used to retrieve an individual authorization rule for access to shared lists.

This light-weight resource is used to manage authorized identities in an existing authorization rule, which include creation, update, retrieval, and delete operations.

6.26.1 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
ruleId	rule identifier: Example: 987654321
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this light-weight resource is defined in the [ResourceRelPath] column in table 5.2.2.19.

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

6.26.2 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 Address Book API, see section 7.

6.26.3 GET

This operation is used for retrieval of authorized identity in an individual authorization rule. It is typically used to verify if it already exists.

6.26.3.1 Example: Retrieve data from an authorization rule (Informative)

6.26.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1/lists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.26.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:listId xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">1234</ab:listId>
```

6.26.4 PUT

This operation is used to add an authorized identity to individual authorization rule.

6.26.4.1 Example: Authorize user by updating an existing authorization rule (Informative)

6.26.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:userId xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">mailto:alice@example.com</ab:userId>
```

6.26.4.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:userId xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">mailto:alice@example.com</ab:userId>
```

6.26.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.26.6 DELETE

This operation removes a previously authorized identity from an authorization rule.

6.26.6.1 Example: Delete an authorized identity (Informative)

6.26.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.26.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT
```

6.27 Resource: Individual member transfer

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/transfer

This resource is used to transfer a member from one list to another list.

6.27.1 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
listId	list identifier. Example: allFriends

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

6.27.2 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 Address Book API, see section 7.

6.27.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

6.27.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

6.27.5 POST

This operation is used to transfer a member from one list to another list. In case of successful transfer, "303 See Other" SHALL be returned, providing a Location header and a resourceReference root element with the location representing the member in the destination list.

6.27.5.1 Example: Transfer a member from one list to another (Informative)

6.27.5.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/members/tel%3A%2B19585550199/transfer
HTTP/1.1Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:memberTransferParameters xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <destination>http://exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts</destination>
</ab:memberTransferParameters>
```

6.27.5.1.2 Response

```
HTTP/1.1 303 See Other
Date: Fri, 10 Dec 2010 12:53:23 GMT
Location:
http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/tel%3A%2B19585550199
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/
tel%3A%2B19585550199</resourceURL>
</common:resourceReference>
```

6.27.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

7. Fault definitions

7.1 Service Exceptions

The following Service Exception codes are defined for the RESTful Address Book API in addition to those in [3GPP 29.199-13].

7.1.1 SVC0240: Key property changes not allowed

Element name	Description
MessageId	SVC0240
Text	Key property changes not allowed: key property %1
Variables	%1 – key property

7.2 Policy Exceptions

The following Policy Exception codes are defined for the RESTful Address Book API in addition to those in [3GPP 29.199-13].

7.2.1 POL0214: Too many resources

Element name	Description
MessageId	POL0214
Text	Maximum number of permitted resources exceeded.
Variables	None

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 OMA-TS-REST_NetAPI_AddressBook-V1_0	03 May 2011	Many	Structural changes to fit the OMA RESTful Network API release. This version inherits the technical content of OMA-TS-ParlayREST_AddressListManagement-V1_0-20110111-C and applies changes according to ARC INP 30R01, 98R02, 155R01,156R01, 167R02, 172R01, 175R01, 186, 187r02, and 167R04 (which also includes the fix equivalent of OMA-REST-M-2011- 2011-0009R01-CR_ALM_ParlayX_fix).
	27 May 2011	Many	Implemented CR: OMA-ARC-REST-NetAPI-2011-0028R03-CR_ALM_TS_Evolution_to_agreed_new_template
	06 Jun 2011	Many	Implemented AI: REST-NetAPI-2011-A025
	22 Jun 2011	Section 5.2.2.7	Implemented CR: OMA-ARC-REST-NetAPI-2011-0054-CR_ALM_TS_fix_value_type_in_Attribute
	28 Jun 2011	Many	Implemented CR OMA-ARC-REST-NetAPI-2011-0053-CR_ALM_address_issue_15_in_ParlayREST_issue_list
	23 Jul 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0126R03-CR_ALM_fix_examples_telURI, OMA-ARC-REST-NetAPI-2011-0148R01-CR_ALM_new_resource_for_member_transfer, editorials
	28 Jul 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0160-CR_ALM_Notif_Channel_changes
	03 Aug 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0168-CR_LW_resource_keys_as_elements_ALM_TS_NetAPI_mirror_of_REST_M_0021, OMA-ARC-REST-NetAPI-2011-0174-CR_ALM_Appendix_C_link_to_section_6 and REST-NetAPI-2011-A090
	11 Aug 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0188-CR_ALM_address_issues_5_and_7_from_ParlayREST_list
	24 Aug 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0190-CR_ALM_resourceURL
	01 Sep 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0200R04-CR_ALM_ACR
	23 Sep 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0214R02-CR_ALM_TS_opaque_vCard, OMA-ARC-REST-NetAPI-2011-0235-CR_ALM_vCard_example
	04 Oct 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0268-CR_ALM_prep_for_CONR
	25 Oct 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0270R02-CR_ALM_TS_changes_for_alignment_with_CAB
	01 Nov 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0333-CR_ALM_implement_CR_326_versioning_blueprint
	09 Nov 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0351-CR_AB_resourceURL_in_examples_and_other_agreed_fixes, OMA-ARC-REST-NetAPI-2011-0353-CR_AB_CR325_resourceURL_blueprint
	16 Nov 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0378R02-CR_AB_TS_prep_for_CONR, OMA-ARC-REST-NetAPI-2011-0379-CR_B_TS_change_SCRs_to_mandatory
	08 Dec 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0381R01-CR_AB_TS_Appendix_G and editorials
	13 Dec 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0441R01-CR_AddressBook_acr_Autho
	10 Jan 2012	Many	Implemented: OMA-ARC-REST-NetAPI-2012-0001R01-CR_AddressBook_CONR_comments_resolution_1
	13 Jan 2012	Many	No content changes – it is a re-issue of the last draft that had many tables formatting issues. For visibility, all previous changes were preserved. Also reversed to Word 2003 format (.doc)

Document Identifier	Date	Sections	Description
	25 Jan2012	Many	Implemented: OMA-ARC-REST-NetAPI-2012-0020R03-CR_AddressBook_resolve_CONR_A019
	26 Jan 2012	Many	Implemented: OMA-ARC-REST-NetAPI-2012-0030R01-CR_AddressBook_Fix_CONR_A035, editorials
	30 Jan 2012	6.12, 6.27	Changed resource URLs to be all bold. Editorial changes
Candidate version OMA-TS-REST_NetAPI_AddressBook-V1_0	07 Feb 2012	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2012-0029-INP_REST_NetAPI_AddressBook_1_0_ERP_and_ETR_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for REST.AB Server

Item	Function	Reference	Requirement
REST-AB-SUPPORT-S-001-M	Support for the RESTful AB API	5, 6	
REST-AB-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-AB-SUPPORT-S-003-M	Support for the JSON request & response format	6	
REST-AB-SUPPORT-S-004-O	Support for the application/x-form-urlencoded format	Appendix C	

B.1.1 SCR for REST.AB.ContactCollection Server

Item	Function	Reference	Requirement
REST-AB-CONTACT-COL-S-001-M	Support contact collection with all of user's contacts	6.1	
REST-AB-CONTACT-COL-S-002-M	This operation retrieves all contacts in the contact collection - GET	6.1.3	

B.1.2 SCR for REST.AB.IndividualContact Server

Item	Function	Reference	Requirement
REST-AB-CONTACT-IND--S-001-M	Support individual contact in contact collection	6.2	
REST-AB-CONTACT-IND-S-002-M	This operation retrieves information of a contact in the contact collection - GET	6.2.3	
REST-AB-CONTACT-IND-S-003-M	This operation adds or updates a contact in the contact collection - PUT	6.2.4	
REST-AB-CONTACT-IND-S-004-M	This operation deletes a contact in the contact collection - DELETE	6.2.6	

B.1.3 SCR for REST.AB.AttributesForAContact Server

Item	Function	Reference	Requirement
REST-AB-ATTRIB-CONTACT-S-001-M	Supports retrieval of all attributes of a contact	6.3	

Item	Function	Reference	Requirement
REST-AB-CONTACT-S-002-M	This operation retrieves all attributes for a contact – GET	6.3.3	

B.1.4 SCR for REST.AB.IndividualAttributeForAContact Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-CONTACT-S-001-M	Support for management (create, update, retrieve and delete) of a contact's single attribute.	6.4	
REST-AB-IND-ATTRIB-CONTACT-S-002-M	This operation returns the value of the attribute for a member – GET	6.4.3	
REST-AB-IND-ATTRIB-CONTACT-S-003-M	This operation creates or updates an attribute for a member – PUT	6.4.4	
REST-AB-IND-ATTRIB-CONTACT-S-004-M	This operation removes an attribute for a member - DELETE	6.4.6	

B.1.5 SCR for REST.AB.Lists Server

Item	Function	Reference	Requirement
REST-AB- MEM-LS-S-001-M	Support for retrieving all lists belonging to a user.	6.5	
REST-AB- MEM-LS-S-002-M	This operation retrieves all lists the user has created - GET	6.5.3	

B.1.6 SCR for REST.AB.IndividualList Server

Item	Function	Reference	Requirement
REST-AB-IND-MEM-LS-001-M	Support for management (create, update, retrieve and delete) of an individual list.	6.6	
REST-AB-IND-MEM-LS-002-M	This operation retrieves the list - GET	6.6.3	
REST-AB-IND-MEM-LS-003-M	This operation creates or updates the entire list - PUT	6.6.4	
REST-AB-IND-MEM-LS-004-M	This operation removes the list including attributes	6.6.6	

Item	Function	Reference	Requirement
	and members - DELETE		

B.1.7 SCR for REST.AB.AttributesForAList Server

Item	Function	Reference	Requirement
REST-AB-ATTRIB-MEM-LS-S-001-O	Support for retrieving all attributes for a given list belonging to a user.	6.7	REST-AB-ATTRIB-MEM-LS-S-002-O
REST-AB-ATTRIB-MEM-LS-S-002-O	This operation returns all attributes for a list - GET	6.7.3	

B.1.8 SCR for REST.AB.IndividualAttributeForAList Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-MEM-LS-MGT-001-O	Support for management (create, update, retrieve and delete) of individual attributes for a list.	6.8	REST-AB-IND-ATTRIB-MEM-LS-MGT-002-O AND REST-AB-IND-ATTRIB-MEM-LS-MGT-003-O AND REST-AB-IND-ATTRIB-MEM-LS-MGT-004-O
REST-AB-IND-ATTRIB-MEM-LS-MGT-002-O	This operation returns the value of the list attribute -GET	6.8.3	
REST-AB-IND-ATTRIB-MEM-LS-MGT-003-O	This operation creates or updates an attribute - PUT	6.8.4	
REST-AB-IND-ATTRIB-MEM-LS-MGT-004-O	This operation deletes an attribute - DELETE	6.8.6	

B.1.9 SCR for REST.AB.MemberInAList Server

Item	Function	Reference	Requirement
REST-AB-MBR-MEM-LS-S-001-O	Supports retrieval of all members of a list belonging to a user	6.9	REST-AB-MBR-MEM-LS-S-002-O
REST-AB-MBR-MEM-LS-S-002-O	This operation retrieves the members in the list - GET	6.9.3	

B.1.10 SCR for REST.AB.IndividualMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-IND-MBR-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of members on individual list.	6.10	REST-AB-IND-MBR-MEM-LS-S-002-O REST-AB-IND-MBR-MEM-LS-S-003-O REST-AB-IND-MBR-MEM-LS-S-

Item	Function	Reference	Requirement
			004-O
REST-AB-IND-MBR-MEM-LS-S-002-O	This operation retrieves a user from a list - GET	6.10.3	
REST-AB-IND-MBR-MEM-LS-S-003-O	This operation creates and updates an entry in the list - PUT	6.10.4	
REST-AB-IND-MBR-MEM-LS-S-004-O	This operation removes the member from the list - DELETE	6.10.6	

B.1.11 SCR for REST.AB.AttributesForAMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-ATTRIB-MBR-MEM-LS-S-001-O	Supports retrieval of all attributes of a member of a list belonging to a user	6.11	REST-AB-ATTRIB-MBR-MEM-LS-S-002-O
REST-AB-ATTRIB-MBR-MEM-LS-S-002-O	This operation returns all attributes for a member – GET	6.11.3	

B.1.12 SCR for REST.AB.IndividualAttributeForAMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of member attributes on individual list.	6.12	REST-AB-IND-ATTRIB-MEM-LS-S-002-O AND REST-AB-IND-ATTRIB-MEM-LS-S-003-O AND REST-AB-IND-ATTRIB-MEM-LS-S-004-O
REST-AB-IND-ATTRIB-MEM-LS-S-002-O	This operation returns the value of the attribute for a member – GET	6.12.3	
REST-AB-IND-ATTRIB-MEM-LS-S-003-O	This operation creates or updates an attribute for a member – PUT	6.12.4	
REST-AB-IND-ATTRIB-MEM-LS-S-004-O	This operation removes an attribute for a member - DELETE	6.12.6	

B.1.13 SCR for REST.AB.ListReferences Server

Item	Function	Reference	Requirement
REST-AB-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested lists.	6.13	REST-AB-MEM-LS-REF-S-002-O

Item	Function	Reference	Requirement
REST-AB-MEM-LS-REF-S-002-O	This operation returns a list of references to other lists. –GET	6.13.3	

B.1.14 SCR for REST.AB.IndivListReference Server

Item	Function	Reference	Requirement
REST-AB-IND-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested lists.	6.14	REST-AB-IND-MEM-LS-REF-S-002-O AND REST-AB-IND-MEM-LS-REF-S-003-O AND REST-AB-IND-MEM-LS-REF-S-004-O
REST-AB-IND-MEM-LS-REF-S-002-O	This operation returns an individual list reference to another list –GET	6.14.3	
REST-AB-IND-MEM-LS-REF-S-003-O	This operation creates or updates an individual reference to another list. –PUT	6.14.4	
REST-AB-IND-MEM-LS-REF-S-004-O	This operation deletes an individual reference to another list. –DELETE	6.14.6	

B.1.15 SCR for REST.AB.List.Subscr Server

Item	Function	Reference	Requirement
REST-AB-SUBSCR-S-001-M	Support subscriptions for address book changes notifications	6.15	
REST-AB-SUBSCR-S-002-M	This operation retrieves a list of subscriptions - GET	6.15.3	
REST-AB-SUBSCR-S-003-M	This operation creates a subscription for list or contact collection change notifications – POST (XML or JSON)	6.15.5	
REST-AB-SUBSCR-S-004-O	This operation creates a subscription for list or contact collection change notifications – POST (application/x-www-form-urlencoded)	C.1	

B.1.16 SCR for REST.AB.Individual.Subscr Server

Item	Function	Reference	Requirement

Item	Function	Reference	Requirement
REST-AB-IND-SUBSCR-S-001-M	Support for managing an individual subscription for list or contact collection change notifications	6.16	
REST-AB-IND-SUBSCR-S-002-M	This operation retrieves an individual subscription for list or contact collection change notifications - GET	6.16.3	
REST-AB-IND-SUBSCR-S-003-M	This operation updates an individual subscription for list or contact collection change notifications - PUT	6.16.4	
REST-AB-IND-SUBSCR-S-004-M	This operation deletes an individual subscription for list or contact collection change notifications - DELETE	6.16.6	

B.1.17 SCR for REST.AB.Notif Server

Item	Function	Reference	Requirement
REST-AB-NOTIF-S-001-M	Support for notifying application about list or contact collection changes	6.17	
REST-AB-NOTIF-S-002-M	This operation notifies the application about list or contact collection changes - POST	6.17.5	

B.1.18 SCR for REST.AB.Shared.ContactCollection Server

Item	Function	Reference	Requirement
REST-AB-SHAREDCONCOLL-S-001-O	Support shared contact collection	6.18	REST-AB-SHAREDCONCOLL-S-002-O
REST-AB-SHAREDCONCOLL-S-002-O	This operation retrieves the collection of shared contacts by one user with another user - GET	6.18.3	

B.1.19 SCR for REST.AB.Individual.Shared.Contact Server

Item	Function	Reference	Requirement
REST-AB-IND-SHAREDCON-S-001-	Support individual shared contact	6.19	REST-AB-IND-SHAREDCON-S-002-O

Item	Function	Reference	Requirement
O			
REST-AB-IND-SHAREDCON-S-002-O	This operation retrieves an individual shared contact by one user with another user - GET	6.19.3	

B.1.20 SCR for REST.AB.Shared.Lists Server

Item	Function	Reference	Requirement
REST-AB-SHAREDLISTS-S-001-O	Support shared lists	6.20	REST-AB-SHAREDLISTS-S-002-O
REST-AB-SHAREDLISTS-S-002-O	This operation retrieves a collection of all shared lists by one user with another user - GET	6.20.3	

B.1.21 SCR for REST.AB.Individual.Shared.List Server

Item	Function	Reference	Requirement
REST-AB-IND-SHAREDLIST-S-001-O	Support individual shared list	6.21	REST-AB-IND-SHAREDLIST-S-002-O
REST-AB-IND-SHAREDLIST-S-002-O	This operation retrieves an individual shared list by one user with another user - GET	6.21.3	

B.1.22 SCR for REST.AB.Members.Ind.Shared.List Server

Item	Function	Reference	Requirement
REST-AB-MEM-IND-SHAREDLIST-S-001-O	Support list of members in individual shared list	6.22	REST-AB-MEM-IND-SHAREDLIST-S-002-O
REST-AB-MEM-IND-SHAREDLIST-S-002-O	This operation retrieves a list of members in an individual shared list by one user with another user - GET	6.22.3	

B.1.23 SCR for REST.AB.Member.Shared.List Server

Item	Function	Reference	Requirement
REST-AB-MEMBER-SHAREDLIST-S-001-O	Support individual member in shared list	6.23	REST-AB-MEMBER-SHAREDLIST-S-002-O

Item	Function	Reference	Requirement
REST-AB-MEMBER-SHAREDLIST-S-002-O	This operation retrieves information of an individual member in a shared list - GET	6.23.3	

B.1.24 SCR for REST.AB.Authorization.RuleList Server

Item	Function	Reference	Requirement
REST-AB-AUTH-RULE-LS-S-001-O	Support authorization rules for sharing lists or contacts	6.24	REST-AB-AUTH-RULE-LS-S-002-O AND REST-AB-AUTH-RULE-LS-S-003-O
REST-AB-AUTH-RULE-LS-S-002-O	This operation retrieves all authorization rules owned by a given user - GET	6.24.3	
REST-AB-AUTH-RULE-LS-S-003-O	This operation creates a new authorization rule – POST (XML or JSON)	6.24.5	
REST-AB-AUTH-RULE-LS-S-004-O	This operation creates a new authorization rule – POST (application/x-www-form-urlencoded)	C.2	

B.1.25 SCR for REST.AB.Indiv.AuthorizationRule Server

Item	Function	Reference	Requirement
REST-AB-IND-AUTH-RULE-S-001-O	Support individual authorization rule	6.25	REST-AB-IND-AUTH-RULE-S-002-O AND REST-AB-IND-AUTH-RULE-S-003-O AND REST-AB-IND-AUTH-RULE-S-004-O
REST-AB-IND-AUTH-RULE-S-002-O	This operation retrieves an individual authorization rule - GET	6.25.3	
REST-AB-IND-AUTH-RULE-S-003-O	This operation updates an individual authorization rule - PUT	6.25.4	
REST-AB-IND-AUTH-RULE-S-004-O	This operation deletes an individual authorization rule - DELETE	6.25.6	

B.1.26 SCR for REST.AB.Indiv.AuthorizationRuleData Server

Item	Function	Reference	Requirement

Item	Function	Reference	Requirement
REST-AB-IND-AUTH-RULE-DATA-S-001-O	Support individual authorization rule data	6.26	REST-AB-IND-AUTH-RULE-DATA-S-002-O AND REST-AB-IND-AUTH-RULE-DATA-S-003-O AND REST-AB-IND-AUTH-RULE-DATA-S-004-O
REST-AB-IND-AUTH-RULE-DATA-S-002-O	This operation retrieves an individual authorization rule specific data - GET	6.26.3	
REST-AB-IND-AUTH-RULE-DATA-S-003-O	This operation updates an individual authorization rule specific data - PUT	6.26.4	
REST-AB-IND-AUTH-RULE-DATA-S-004-O	This operation deletes an individual authorization rule specific data - DELETE	6.26.6	

B.1.27 SCR for REST.AB.IndividualMemberTransfer Server

Item	Function	Reference	Requirement
REST-AB-IND-MEM-TRANS-S-001-M	Supports for transferring a member to another list	6.27	
REST-AB-IND-MEM-TRANS-S-002-M	Transferring a member to another list – POST (XML or JSON)	6.27.5	
REST-AB-IND-MEM-TRANS-S-003-O	Transferring a member to another list – POST (application/x-www-form-urlencoded)	C.3	

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

This section defines a format for RESTful Address Book API requests where the body of the request is encoded using the application/x-www-form-urlencoded MIME type.

Note: only the request body is encoded as application/x-www-form-urlencoded, the response is still encoded as XML or JSON depending on the preference of the client and the capabilities of the server. Names and values MUST follow the application/x-www-form-urlencoded character escaping rules from [W3C_URLENC].

The encoding is defined below for the following Address Book REST operations which are based on POST requests:

- Create a subscription for list changes notification
- Create an authorization rule
- Transfer a member from one list to another list

C.1 Create a subscription for list changes notifications

This operation is used to create a new subscription for list changes notifications, see section 6.15.5.

This REST operation is used by the application to start the list changes notifications. It MUST use the HTTP POST operation. If the operation was successful, it returns an HTTP Status of “201 Created”.

The notifyURL either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]).

The following parameters are defined:

Name	Type/Values	Optional	Description
anyContacts	xsd:boolean	Choice	Indicates a subscription for changes about any contacts in the 'ContactCollection'.
listId	xsd:anyURI	Choice	Indicates a subscription for changes about the specified list.
notifyURL	xsd:anyURI	No	Notification endpoint definition. For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.15 and 6.15.5.
callbackData	xsd: string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
duration	xsd:int	Yes	Subscription duration in seconds. Server would expire subscription after specified number of seconds after subscription creation. If not specified – default value assigned by the server.
notificationFormat	xsd: string	Yes	Default: XML Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}

Note: "choice" indicates an optional selection of either anyContacts or listId.

C.1.1 Example using 'tel' URI (Informative)

C.1.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/subscriptions/abChanges HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

listId=1234&
notifyURL=http://application.example.com/notifications/lists/1234&
callbackData=12345&
```

```
clientCorrelator=123&
applicationTag=myApp&
duration=7200&
```

C.1.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

C.1.2 Example using ‘acr’ URI (Informative)

C.1.2.1 Request

```
POST /exampleAPI/addressbook/v1/acr%3Apseudonym123/lists/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

listId=1234&
notifyURL=http://application.example.com/notifications/lists/1234&
callbackData=12345&
clientCorrelator=123&
applicationTag=myApp&
duration=7200&
```

C.1.2.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ab:abChangesSubscription xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <listId>1234</listId>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/lists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321
</resourceURL>
</ab:abChangesSubscription>
```

C.2 Create an authorization rule

This operation is used to create an authorization rule, see section 6.24.5. It MUST use the HTTP POST operation. If the operation was successful, it returns an HTTP Status of “201 Created”.

The following parameters are defined:

Name	Type/Values	Optional	Description
ruleName	xsd:ID	No	A name associated with the rule. It is a key property of the rule and when included in the light-weight resource URL then it SHALL NOT be altered.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applToContacts	xsd:anyURI [0..unbounded]	Choice1	The rule applies to the list of specified 'contactId' that are in the 'ContactCollection'.
applToAllContacts	xsd:boolean	Choice1	The rule applies to all contacts.
applToLists	xsd:anyURI [0..unbounded]	Choice2	The rule applies to the specified list of 'listId'.
applToAllLists	xsd:boolean	Choice2	The rule applies to all lists.
applToListsAndContacts	xsd:anyURI [0..unbounded]	Choice3	The rule applies to the specified list. The rule applies also to contacts that are referenced from the specified list.
applToAllListsAndContacts	xsd:boolean	Choice3	The rule applies for all lists and referenced contacts.
authorizedToUserId	xsd:string [0..unbounded]	Choice4	User is authorized to access a resource for which the rule is applicable to.
authorizedToAllSharedIdentities	xsd:boolean	Choice4	All contacts with shared identities are authorized to access a resource for which the rule is applicable to.
authorizedToListId	xsd:anyURI [0..unbounded]	Choice4	Users within the list are authorized to access a resource for which the rule is applicable to.
authorizedToAllLists	xsd:boolean	Choice4	Users within all lists are authorized to access a resource for which the rule is applicable to.
authorizedToExternalList	xsd:anyURI [0..unbounded]	Choice4	Users within the list (owned by another user) are authorized to access a resource for which the rule is applicable to.
authorizedToDomainName	xsd:string [0..unbounded]	Choice4	Users within the domain are authorized to access a resource for which the rule is applicable to.
authorizedToOtherUser	xsd:boolean	Choice4	Any user is authorized to access a resource for which the rule is applicable to.
listFilter	xsd:string [0..unbounded]	Yes	Filter for list attributes (section 6.5) the retrieving user is allowed to see. An empty filter means that the user has access to all attributes.
indivFilter	xsd:string [0..unbounded]	Yes	Filter for member and contact attributes (section 6.1 and 6.9) the retrieving user is allowed to see. An empty filter means that the user has access to all attributes.

Note:

- "choice1" indicates an optional selection of either applicableToContacts or applicableToAllContacts
- "choice2" indicates an optional selection of either applicableToLists or applicableToAllLists

- "choice3" indicates an optional selection of either applicableToListsAndContacts or applicableToAllListsAndContacts
- "choice4" indicates an optional selection of either authorizedToUserId or authorizedToAllSharedIdentities or authorizedToListId or authorizedToAllLists or authorizedTopplicableToContacts or applicableToAllContacts

C.2.1 Example (Informative)

C.2.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

ruleName=combinedRule&
clientCorrelator=135&
applicationTag=myApp&
applToListsAndContacts=5678&
authorizedToListId=5678
```

C.2.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:rule xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <ruleName>combinedRule</ruleName>
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToListsAndContacts>
    <listId>5678</listId>
  </applToListsAndContacts>
  <authorizedTo>
    <listId>5678</listId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3</resourceURL>
</ab:rule>
```

C.3 Transfer a member from one list to another

This operation is used to transfer a member from one list to another, see section 6.27.5.

The following parameters are defined:

Name	Type/Values	Optional	Description
destination	xsd:anyURI	No	Contains a relative URI representing the list where the member is to be transferred.

C.3.1 Example: (Informative)

C.3.1.1 Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/members/tel%3A%2B19585550199/transfer HTTP/1.1

Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

destination=
http%3A%2F%2Fexample.com%2FexampleAPI%2F1%2Faddressbook%2Ftel%3A%2B19585550100%2Flists%2FblockedContacts
```

C.3.1.2 Response

```
HTTP/1.1 303 See Other
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location:
http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/tel%3A%2B19585550199
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/
tel%3A%2B19585550199</resourceURL>
</common:resourceReference>
```

Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a lightweight, 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 invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC 4627].

The following examples show the request and response for various operations using a JSON binding. 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 Retrieve all contacts with all attributes (default) (section 6.1.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"contactCollection": {
   "contact": [
      {
         "attributeList": {"attribute": [
            {
               "name": "display-name",
               "value": "Alice"
            },
            {
               "name": "cellphone",
               "value": "tel:+19585550109"
            },
            {
               "name": "state",
               "value": "California"
            }
         ],
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes"
         },
         "contactId": "alice",
         "link": {
            "href":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122",
```

```
            "rel": "Member"
         },
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice"
      },
      {
         "attributeList": {"attribute": [
            {
               "name": "display-name",
               "value": "Sam"
            },
            {
               "name": "cellphone",
               "value": "tel:+19585550108"
            },
            {
               "name": "state",
               "value": "New Jersey"
            }
         ],
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam/attributes"
         },
         "contactId": "sam",
         "link": [
            {
               "href":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550121",
               "rel": "Member"
            },
            {
               "href":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/tel%3A%2B19585550121",
               "rel": "Member"
            }
         ],
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/sam",
         "sharedIdentity": {"sharedId": "tel:+19585550121"}
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts"
}}
```


D.2 Retrieve a contact with all attributes (default) (section 6.2.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"contact": {
    "attributeList": {"attribute": [
        {
            "name": "display-name",
            "value": "Alice"
        },
        {
            "name": "cellphone",
            "value": "tel:+19585550109"
        },
        {
            "name": "state",
            "value": "California"
        }
    ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes"
    },
    "contactId": "alice",
    "link": {
        "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122",
        "rel": "Member"
    },
    "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice"
}}
```

D.3 Retrieve a contact with only vCard2.1 as selected attribute (section 6.2.3.3)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"contact": {
    "attributeList": {"attribute": [
        {
            "name": "vCard2.1",
            "objectValue":
"QkVHSU46VkNBUkQNClZFUlNJT046Mi4xDQpOOkd1bXA7Rm9ycmVzdA0KRk46Rm9ycmVzdCBHdW1wDQpPUkc6QnViYmEgR3Vtc
```

CBTaHJpbXAgQ28uDQpUSVRMRTpTaHJpbXAgTWFuDQpURUw7V09SSztWT0lDRTooMTExKSA1NTUtMTIxMg0KVEVMO0hPTUU7
Vk9JQ0U6KDQwNCkgNTU1LTEyMTINCkFEUjtXT1JLOjs7MTAwIFdhdGVycyBFZGdlO0JheXRvd247TEE7MzAzMTQ7VW5pdGVkIFN
0YXRlcyBvZiBBbWVyaWNhDQpMQUJFTDtXT1JLO0VOQ09ESU5HPVFVT1RFRC1QUklOVEFCTEU6MTAwIFdhdGVycyBFZGdlPTBE
PTBBQmF5dG93biwgTEEgMzAzMTQ9MEQ9MEFVbml0ZWQgU3RhdGVzIG9mIEFtZXJpY2ENCkFEUjtIT01FOjs7NDIgUGxhbnRhdGlv
biBTdC47QmF5dG93bjtMQTszMDMxNDtVbml0ZWQgU3RhdGVzIG9mIEFtZXJpY2ENCkxBQkVMO0hPTUU7RU5DT0RJTkc9UVVPVE
VELVBSSU5UQUJMRTo0MiBQbGFudGF0aW9uIFN0Lj0wRD0wQUJheXRvd24sIExBIDMwMzE0PTBEPTBBVW5pdGVkIFN0YXRlcyBv
ZiBBbWVyaWNhDQpFTUFJTDtQUkVGO0lOVEVSTkVUOmZvcnJlc3RndW1wQGV4YW1wbGUuY29tDQpSRVY6MjAwODA0MjRUMTk
1MjQzWg0KRU5EOlZDQVJE"
      }
   ]
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes"
   },
   "contactId": "alice",
   "link": {
      "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550122",
      "rel": "Member"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice"
}}

D.4 Create a new contact (section 6.2.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria  HTTP/1.1
Content-Type: application/json
Accept: application/json
Content-Length : nnnn
Host: example.com

{"contact": {
   "attributeList": {"attribute": {
      "name": "cellphone",
      "value": "tel:+19585550106"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes"
   },
   "contactId": "maria",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria
Content-Type: application/json
Content-Length: nnnn
```

{"contact": {
"attributeList": {"attribute": {
"name": "cellphone",
"value": "tel:+19585550106"
},
"resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes"
},
"contactId": "maria",
"resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria"
}}

D.5 Retrieve all attributes of a contact (section 6.3.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attributeList": {
"attribute": {
"name": "cellphone",
"value": "tel:+19585550106"
},
"resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/maria/attributes"
}}
```

D.6 Retrieve a contact’s attribute (section 6.4.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
```

```
Content-Length: nnnn

{"attribute": {
   "name": "cellphone",
   "value": "tel:+19585550109"
}}
```

D.7 Create or update a contact's attribute (section 6.4.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/alice/attributes/married  HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length : nnnn

{"attribute": {
   "name": "married",
   "value": "true"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "married",
   "value": "true"
}}
```

D.8 Retrieve specific lists attributes and specific member attributes (section 6.5.3.3)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists?listFilter=display-name&indivFilter=display-name&indivFilter=phone
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"listCollection": {"list": [
   {
      "attributeList": {"attribute": {
         "name": "display-name",
         "value": "Friends"
      },
      "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes"
      },
      "category": "URIList",
      "memberCollection": {"member": {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Alice"
         },
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/
members/sip%3Aalice@example.com/attributes"
         },
         "memberId": "sip:alice@example.com",
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com"
         },
      "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members"
      },
      "listId": "1234",
      "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234",
      "sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
   },
   {
      "attributeList": {"attribute": {
         "name": "display-name",
         "value": "Family"
      },
      "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/attributes"
      },
      "memberCollection": {"member": [
         {
            "attributeList": {"attribute": {
               "name": "display-name",
               "value": "Wife"
            },
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/mailto%3Aliza@example.com/attributes"
            },
            "memberId": "mailto:liza@example.com",
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aliza@example.com"
         },
         {
            "attributeList": {"attribute": [
               {
```

```
                    "name": "display-name",
                    "value": "Son"
                },
                {
                    "name": "phone",
                    "value": "tel:+19585550105"
                }
            ],
            "resourceURL":
http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/mailto%3Aserezha@example.com/
attributes
            },
            "memberId": "mailto:serezha@example.com",
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members/mailto%3Aserezha@example.com"
          }
       ],
       "resourceURL":     http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678/members
},
       "listId": "5678",
       "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678"
   }
]},
"resourceURL": http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists
}
```


D.9 Retrieve a list with all attributes and all member attributes (default) (section 6.6.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"list": {
   "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Friends"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes"
   },
```

```
   "memberCollection": {"member": {
      "attributeList": {"attribute": {
         "name": "display-name",
         "value": "Alice"
      },
      "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com/attributes"
      },
      "memberId": "sip:alice@example.com",
      "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com"
   }},
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234"
}}
```


D.10 Retrieve a non existing list (section 6.6.3.4)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/4321 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"requestError": {
   "link": {
      "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/4321",
      "rel": "List"
   },
   "serviceException": {
      "messageId": "SVC0002",
      "text": "Invalid input value for message part %1",
      "variables": "listId"
   }
}}
```

D.11 Create a list (section 6.6.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"list": {
   "memberCollection": {
   "member": {
   "memberId": mailto:alice@example.com,
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members/mailto%3Aalice@example.com"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members"
   },
   "listId": "bobPublic",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic
Content-Type: application/json
Content-Length: nnnn

{"list": {
   "memberCollection": {
   "member": {"memberId": mailto:alice@example.com,
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members/mailto%3Aalice@example.com"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic/members"
   },
   "listId": "bobPublic",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/bobPublic"
}}
```

D.12 Retrieve all attributes for a list belonging to Bob (section 6.7.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attributeList": {
   "attribute": [
      {
         "name": "display-name",
         "value": "Friends"
      },
      {
         "name": "date-created",
         "value": "Fri, 10 Dec 2010 11:28:34 GMT"
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes"
}}
```

D.13 Retrieve an attribute value (section 6.8.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/date-created HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "date-created",
   "value": "Fri, 10 Dec 2010 11:28:34 GMT"
}}
```

D.14 Create an attribute (section 6.8.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/attributes/display-name  HTTP/1.1
Accept: application/json
Host: example.com
```

```
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "display-name",
   "value": "Close friends"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://{serverRoot}/addressbook/{apiVersion}/tel%3A%2B19585550100/lists/{listId}
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "display-name",
   "value": "Close friends"
}}
```

D.15 Retrieve all members for a list belonging to a user with all attributes (default) (section 6.9.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"memberCollection": {
   "member": [
      {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Alice"
         },
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com/
attributes"
         },
         "memberId": "mailto:alice@example.com",
```

```
          "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com"
       },
       {
          "attributeList": {"attribute": {
             "name": "display-name",
             "value": "Bob"
          },
          "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550124/attributes"
          },
          "memberId": "tel:+19585550124",
          "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550124"
       }
   ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members"
}}
```


D.16 Retrieve a member and selected attributes (section 6.10.3.2)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/sip%3Aalice@example.com?indivFilter=cellphone
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"member": {
   "attributeList": {"attribute": {
      "name": "cellphone",
      "value": "tel:+19585550155"
   },
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com"
   },
   "memberId": "sip:alice@example.com",
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com"
}}
```

D.17 Create a member without a link to the contact (section 6.10.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com   HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length : nnnn

{"member": {
   "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Ole"
   },
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com/attributes"
   },
   "memberId": "mailto:ole@example.com",
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3A:ole@example.com
Content-Type: application/json
Content-Length: nnnn

{"member": {
   "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Ole"
   },
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com/attributes"
   },
   "memberId": "mailto:ole@example.com",
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aole@example.com"
}}
```

D.18 Retrieve all attributes of a member of a list (section 6.11.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/mailto%3Aalice@example.com/attributes HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"attributeList": {
   "attribute": [
      {
         "name": "display-name",
         "value": "Alice"
      },
      {
         "name": "cellphone",
         "value": "tel:+19585550155"
      }
   ],
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com/
attributes"
}}
```

D.19 Retrieve a member's non existing attribute (section 6.12.3.2)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/pet
 Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 404 Not Found
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

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

D.20 Create a member's attribute (section 6.12.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/attributes/born  HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "born",
   "value": "1984"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/6789/members/sip%3Aalice@example.com/
attributes/born
Content-Type: application/json
Content-Length: nnnn

{"attribute": {
   "name": "born",
   "value": "1984"
}}
```

D.21 Retrieve all list references in a list (section 6.13.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"listReferenceCollection": {
   "link": [
      {
```

```
            "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234",
            "rel": "List"
        },
        {
            "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678",
            "rel": "List"
        }
    ],
    "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/friends/listReferences"
}}
```

D.22 Retrieve a list reference (section 6.14.3.1)

Request:

```
GET
/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/http%3A//example.com/exampleAPI/addressbook/
v1/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"link": {
    "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/1234",
    "rel": "List"
}}
```

D.23 Create a list reference (section 6.14.4.1)

Request:

```
PUT
/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/http%3A//example.com/exampleAPI/addressbook/v
1/tel%3A%2B19585550100/lists/5678 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: example.com
Content-Length: nnnn
```

```
{"link": {
   "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678",
   "rel": "List"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/listReferences/
http%3A//example.com/exampleAPI/addressbook/v1/tel%3A%2B 19585550100/lists/5678
Content-Type: application/json
Content-Length: nnnn

{"link": {
   "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/5678",
   "rel": "List"
}}
```

D.24 Retrieve all list changes subscriptions (section 6.15.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscriptionCollection": {
   "abChangesSubscription": [
      {
         "applicationTag": "myApp",
         "callbackReference": {
            "callbackData": "12345",
            "notifyURL": "http://application.example.com/notifications/lists/1234"
         },
         "clientCorrelator": "123",
         "duration": "1958",
         "listId": "1234",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"
      },
      {
         "anyContacts": null,
```

```
        "applicationTag": "myApp",
        "callbackReference": {
           "callbackData": "54321",
           "notifyURL": "http://application.example.com/notifications/contacts"
        },
        "clientCorrelator": "456",
        "duration": "2641",
        "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/123456789"
     }
  ],
  "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges"
}}
```


D.25 Create new subscription for list changes notification using ‘tel’ URI (section 6.15.5.1)

Request

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/lists/1234"
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234"
}}
```

Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": http://application.example.com/notifications/lists/1234
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"
}}
```

D.26 Create new subscription for list changes notification using ‘acr’ URI (section 6.15.5.2)

Request

```
POST /exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/lists/1234"
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234"
}}
```

Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": http://application.example.com/notifications/lists/1234
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/acr%3Apseudonym123/subscriptions/abChanges/987654321"
}}
```

D.27 Retrieve individual list changes subscription (section 6.16.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/lists/1234"
   },
   "clientCorrelator": "123",
   "duration": "6817",
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"
}}
```

D.28 Update/Extend duration of subscription (section 6.16.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/lists/1234"
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"abChangesSubscription": {
```

```
   "applicationTag": "myApp",
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/lists/1234"
   },
   "clientCorrelator": "123",
   "duration": "7200",
   "listId": "1234",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/987654321"
}}
```

D.29 Notification for contact change subscription (section 6.17.5.2)

Request:

```
POST /notifications/contacts HTTP/1.1
Accept: application/json
Host: application.example.com
Content-Type: application/json
Content-Length: nnnn

{"abChangesNotification": {
   "callbackData": "54321",
   "duration": "4217",
   "link": [
      {
         "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/contacts/ole",
         "rel": "Contact"
      },
      {
         "href": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/subscriptions/abChanges/123456789",
         "rel": "AbChangesSubscription"
      }
   ],
   "resourceStatus": "Active"
} }
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

D.30 Retrieve contacts shared by a user identified by otherUserId with all contact attributes (section 6.18.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B 19585550105/shared/tel%3A%2B19585550100/contacts HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"contactCollection": {
   "contact": [
      {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Alice"
         },
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
sip%3Aalice@example.com/attributes"
         },
         "contactId": "alice",
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
sip%3Aalice@example.com",
         "sharedIdentity": {"sharedId": "sip:alice@example.com"}
      },
      {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Ole"
         },
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
mailto%3Aole@example.com/attributes"
},
         "contactId": "ole",
         "link": {
            "href":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/shared/lists/6789/members/mailto%3Aole@example.com",
            "rel": "Member"
         },
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/
mailto%3Aole@example.com",
         "sharedIdentity": {"sharedId": "mailto:ole@example.com"}
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts"
}}
```

D.31 Retrieve information about individual shared contact using contactId (section 6.19.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/contacts/alice HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"contact": {
   "attributeList": {"attribute": [
      {
         "name": "display-name",
         "value": "Alice"
      },
      {
         "name": "cellphone",
         "value": "tel:+19585550109"
      },
      {
         "name": "state",
         "value": "California"
      }
   ].
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/alice/attributes"
   },
   "contactId": "alice",
   "link": {
      "href":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
tel%3A%2B19585550122",
      "rel": "Member"
   },
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/alice"
}}
```

D.32 Retrieve lists shared by a user identified by otherUserId without lists attributes, but with selected member attributes (section 6.20.3.2)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists?listFilter=~noAttr& indivFilter=phone
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"listCollection": {"list": [
   {
      "category": "URIList",
      "memberCollection": {"member": {
         "memberId": "sip:alice@example.com",
         "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com"
      },
      "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members"
      },
      "listId": "1234",
      "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234",
      "sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
   },
   {
      "memberCollection": {"member": [
         {
            "memberId": "mailto:liza@example.com",
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aliza@example.com"
         },
         {
            "attributeList": {"attribute": {
               "name": "phone",
               "value": "tel:+19585550105"
            },
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com/attributes"
            },
            "memberId": "mailto:serezha@example.com",
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/members/
mailto%3Aserezha@example.com"
         }
      ],
```

```
            "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678/members"
        },
        "listId": "5678",
        "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/5678"
    }
]},
        "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists"
}
```


D.33 Retrieve an individual list shared by a user identified by otherUserId, with all list attributes and all member attributes (section 6.21.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:


```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"list": {
    "attributeList": {"attribute": {
        "name": "display-name",
        "value": "Friends"
    },
    "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/attributes"
    },
    "category": "URIList",
    "memberCollection": {"member": {
        "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Alice"
        },
        "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes"
        },
        "memberId": "sip:alice@example.com",
        "resourceURL":
```

"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com"
}},
"listId": "1234",
"resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234",
"sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
}}

D.34 Retrieve all members for a shared list with all attributes (default) (section 6.22.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"memberCollection": {
   "member": [
      {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Alice"
         },
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/
tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com/
attributes"
         },
         "memberId": "mailto:alice@example.com",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/
tel%3A%2B19585550100/lists/1234/members/mailto%3Aalice@example.com"
      },
      {
         "attributeList": {"attribute": {
            "name": "display-name",
            "value": "Bob"
         },
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/
tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550124/attributes"
         },
         "memberId": "tel:+19585550124",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/
tel%3A%2B19585550100/lists/1234/members/tel%3A%2B19585550124"
      }
```

```
   ],
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/
tel%3A%2B19585550100/lists/1234/members"
}}
```

D.35 Retrieve selected member attributes about user identified as memberId from the list shared by a user identified by otherUserId (section 6.23.3.2)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com?indivFilter=cellphone HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:53:23 GMT
Content-Type: application/json
Content-Length: nnnn

{"member": {
   "attributeList": {"attribute": {
      "name": "cellphone",
      "value": "tel:+19585550105"
   },
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com/attributes"
},
   "memberId": "sip:alice@example.com",
   "resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550105/shared/tel%3A%2B19585550100/lists/1234/members/
sip%3Aalice@example.com"
}}
```

D.36 Retrieve all authorization rules (section 6.24.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"ruleList": {
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules",
   "rule": [
      {
         "applToContacts": {"allContacts": null},
         "applicationTag": "myApp",
         "authorizedTo": {"listId": "1234"},
         "clientCorrelator": "123",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1",
         "ruleName": "allowAllContacts"
      },
      {
         "applToContacts": {"contactId": "bob"},
         "applicationTag": "myApp",
         "authorizedTo": {"allSharedIdentities": null},
         "clientCorrelator": "456",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2",
         "ruleName": "allowOwnContact"
      },
      {
         "applToLists": {"listId": "1234"},
         "applicationTag": "myApp",
         "authorizedTo": {"listId": "1234"},
         "clientCorrelator": "789",
         "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/autorizationRules/rule3",
         "ruleName": "allowFriendsList"
      }
   ]
}}
```

D.37 Create new rule for list and referenced contacts (section 6.24.5.1)

Request:

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "applToListsAndContacts": {"listId": "5678"},
   "applicationTag": "myApp",
```

```
   "authorizedTo": {"listId": "5678"},
   "clientCorrelator": "135",
   "ruleName": "combinedRule"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010  21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "applToListsAndContacts": {"listId": "5678"},
   "applicationTag": "myApp",
   "authorizedTo": {"listId": "5678"},
   "clientCorrelator": "135",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule3",
   "ruleName": "combinedRule"
}}
```

D.38 Retrieve an authorization rule (section 6.25.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "applToContacts": {"allContacts": null},
   "applicationTag": "myApp",
   "authorizedTo": {"listId": "1234"},
   "clientCorrelator": "123",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1",
   "ruleName": "allowAllContacts"
}}
```

D.39 Update an existing authorization rule (section 6.25.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length : nnnn

{"rule": {
   "applToContacts": {"allContacts": null},
   "applicationTag": "myApp",
   "authorizedTo": {"listId": [
      "1234",
      "5678"
   ]},
   "clientCorrelator": "123",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1",
   "ruleName": "allowAllContacts"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"rule": {
   "applToContacts": {"allContacts": null},
   "applicationTag": "myApp",
   "authorizedTo": {"listId": [
      "1234",
      "5678"
   ]},
   "clientCorrelator": "123",
   "resourceURL": "http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1",
   "ruleName": "allowAllContacts"
}}
```

D.40 Retrieve data from an authorization rule (section 6.26.3.1)

Request:

```
GET /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule1/lists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:55:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"listId": "1234"}
```

D.41 Authorize user by updating an existing authorization rule (section 6.26.4.1)

Request:

```
PUT /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length : nnnn

{"userId": "mailto:alice@example.com"}
```

Response:

```
HTTP/1.1 201 Created
Date: Fri, 10 Dec 2010 12:55:59 GMT
Location: /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com
Content-Type: application/json
Content-Length: nnnn

{"userId": "mailto:alice@example.com"}
```

D.42 Delete an authorized identity (section 6.26.6.1)

Request:

```
DELETE /exampleAPI/addressbook/v1/tel%3A%2B19585550100/authorizationRules/rule2/users/mailto%3Aalice@example.com
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content
```

```
Date: Fri, 10 Dec 2010 12:55:59 GMT
```

D.43 Transfer a member from one list to another (section 6.27.5.1)

Request:

```
POST /exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/allFriends/members/tel%3A%2B19585550200/transfer HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"memberTransferParameters": {
"destination": "http://exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts"
}}
```

Response:

```
HTTP/1.1 303 See Other
Date: Fri, 10 Dec 2010 12:55:59 GMT
Location:
http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/tel%3A%2B19585550199
Content-Type: application/json
Content-Length: nnnn

{"resourceReference": {
"resourceURL":
"http://example.com/exampleAPI/addressbook/v1/tel%3A%2B19585550100/lists/blockedContacts/members/tel%3A%2B19585550199"
}}
```

Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/methods and Parlay X [3GPP 29.199-13] equivalent operations.

REST Resource	REST Method	REST Section reference	Parlay X equivalent operation
Lists	GET	6.5.3	getOwnersGroups
Individual list	DELETE	6.6.6	deleteGroup
Attributes for a list	GET	6.7.3	queryGroupAttributes
Individual attribute for a list	DELETE	6.8.6	deleteGroupAttribute
Members in a list	GET	6.9.3	queryMembers
Individual member in a list	DELETE	6.10.6	deleteMember
Attributes for a member in a list	GET	6.11.3	queryGroupMemberAttributes
Individual attribute for a member in a list	DELETE	6.12.6	deleteGroupMemberAttribute

Table 1: Parlay X operations mapping

Appendix F. Light-weight resources (Informative)

The following table lists all Address Book data structure elements that can be accessed individually as light-weight resources. For each light-weight resource there are listed: corresponding root element name, root element type and [ResourceRelPath] string.

Type of light-weight resources (and references to data structures)	Element/attribute that can be accessed as light-weight resource	Root element name for the light-weight resource	Root element type for the light-weight resource	[ResourceRelPath] string that needs to be appended to the corresponding heavy-weight resource URL
Attributes (5.2.2.6)	attribute	attribute	Attribute	{name}
List references (5.2.2.10)	link	link	common:Link	{href}
Authorization (5.2.2.19)	userId	userId	xsd:anyURI	users/{userId}
	listId	listId	xsd:anyURI	lists/{listId}
	link	link	common:Link	externalLists/{href}
	domainName	domainName	xsd:string	domains/{domainName}

Table 2: Light-weight resources for Address Book

Note: When appending [ResourceRelPath] string to its heavy-weight resource URL, all variables within curly brackets "{}" such as: serviceId, version, deviceId, watcherUserId, contactListId, and domainName have to be replaced by their real values.

Appendix G. Authorization aspects (Normative)

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

G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Address Book API MAY support the authorization framework defined in [Autho4API_10].

A RESTful Address Book 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 Address Book 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_addressbook.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_addressbook.contacts	Provide access to all defined operations on contacts.	No
oma_rest_addressbook.lists	Provide access to all defined operations on lists.	No
oma_rest_addressbook.subscr	Provide access to all defined operations on subscriptions.	No
oma_rest_addressbook.shared	Provide access to all defined operations on shared contacts/lists.	No
oma_rest_addressbook.rules	Provide access to all defined operations on authorization rules for shared contacts/lists.	No

Table 3: Scope values for RESTful Address Book API

G.1.1.2 Downscoping

In the case where the client requests authorization for “oma_rest_addressbook.all_{apiVersion}” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma_rest_addressbook.contact”
- “oma_rest_addressbook.list”
- “oma_rest_addressbook.subscr”

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 Address Book API map to the REST resources and methods of this API. In these tables, the root “oma_rest_addressbook.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of contacts	/{userId}/contacts	6.1	**all_{apiVersion}** or **contact**	n/a	n/a	n/a
Individual contact	/{userId}/contacts/{contactId}	6.2	**all_{apiVersion}** or **contact**	**all_{apiVersion}** or **contact**	n/a	**all_{apiVersion}** or **contact**
Attributes for a contact	/{userId}/contacts/{contactId}/attributes	6.3	**all_{apiVersion}** or **contact**	n/a	n/a	n/a
Individual attribute for a contact	/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]	6.4	**all_{apiVersion}** or **contact**	**all_{apiVersion}** or **contact**	n/a	**all_{apiVersion}** or **contact**
Collection of shared contacts	/{userId}/shared/{otherUserId}/contacts	6.18	**all_{apiVersion}** or **contact**	n/a	n/a	n/a
Individual shared contact	/{userId}/shared/{otherUserId}/contacts/{sharedIdentity}	6.19	**all_{apiVersion}** or **contact**	n/a	n/a	n/a

Table 4: Required scope values for: managing contacts

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Lists	/{userId}/lists	6.5	**all_{apiVersion}** or **list**	n/a	n/a	n/a
Individual list	/{userId}/lists/{listId}	6.6	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	**all_{apiVersion}** or **list**

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Attributes for a list	/{userId}/lists/{listId}/attributes	6.7	**all_{apiVersion}** or **list**	n/a	n/a	n/a
Individual attribute for a list	/{userId}/lists/{listId}/attributes/[ResourceRelPath]	6.8	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	**all_{apiVersion}** or **list**
Members in a list	/{userId}/lists/{listId}/members	6.9	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	n/a
Individual member in a list	/{userId}/lists/{listId}/members/{memberId}	6.10	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	**all_{apiVersion}** or **list**
Attributes for a member in a list	/{userId}/lists/{listId}/members/{memberId}/attributes	6.11	**all_{apiVersion}** or **list**	n/a	n/a	n/a
Individual attribute for a member in a list	/{userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]	6.12	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	**all_{apiVersion}** or **list**
Individual member transfer	/{userId}/lists/{listId}/members/{memberId}/transfer	6.27	n/a	n/a	**all_{apiVersion}** or **list**	n/a
List references	/{userId}/lists/{listId}/ listReferences	6.13	**all_{apiVersion}** or **list**	n/a	n/a	n/a
Individual list reference	/{userId}/lists/{listId}/ listReferences/[ResourceRelPath]	6.14	**all_{apiVersion}** or **list**	**all_{apiVersion}** or **list**	n/a	**all_{apiVersion}** or **list**
Collection of shared lists	/{userId}/shared/{otherUserId}/lists	6.20	**all_{apiVersion}** or **list**	n/a	n/a	n/a

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Individual shared list	/{userId}/shared/{otherUserId}/lists/{sharedListIdentity}	6.21	**all_{apiVersion}** or **list**	n/a	n/a	n/a
Individual member information from shared list	/{userId}/shared/lists/{otherUserId}/{sharedListIdentity}/{memberId}	6.23	**all_{apiVersion}** or **list**	n/a	n/a	n/a

Table 5: Required scope values for: managing lists and members in a list

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Address Book changes subscriptions	/{userId}/subscriptions/abChanges	6.15	**all_{apiVersion}** or **subscr**	n/a	**all_{apiVersion}** or **subscr**	n/a
Individual address book changes subscription	/{userId/ subscriptions/ abChanges/{subscriptionId}	6.16	**all_{apiVersion}** or **subscr**	**all_{apiVersion}** or **subscr**	n/a	**all_{apiVersion}** or **subscr**

Table 5: Required scope values for: managing subscriptions to notifications about contacts and/or lists changes

Resource	URL Base URL: http://{serverRoot}/addressbook/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Authorization rules	/{userId}/authorizationRules	6.24	**all_{apiVersion}** or **contact** or **list**	n/a	**all_{apiVersion}** or **contact** or **list**	n/a
Individual authorization rule	/{userId}/authorizationRules/{ruleId}	6.25	**all_{apiVersion}** or **contact** or **list**	**all_{apiVersion}** or **contact** or **list**	**all_{apiVersion}** or **contact** or **list**	**all_{apiVersion}** or **contact** or **list**
Individual authorization rule data	/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]	6.26	**all_{apiVersion}** or **contact** or **list**	**all_{apiVersion}** or **contact** or **list**	n/a	**all_{apiVersion}** or **contact** or **list**

Table 6: Required scope values for: managing authorization rules for shared contacts/lists/members

G.1.2 Use of 'acr:Authorization'

This section specifies the use of 'acr:Authorization' in place of an end user identifier in a resource URL path.

An 'acr' URI of the form 'acr:Authorization', where 'Authorization' 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:Authorization' in a resource URL in place of the {endUserId} resource URL variable in the resource URL path, when the RESTful AddressBook API is used in combination with [Autho4API_10].

In the case the RESTful Payment supports [Autho4API_10], the server:

- SHALL accept 'acr:Authorization' as a valid value for the resource URL variable {endUserId}.
- SHALL conform to [REST_Common_TS] section 5.8.1.1 regarding the processing of 'acr:Authorization'.

Appendix H. Overview of data types (Informative)

The section provides an overview of the relation between the data types used for contacts, members and lists.

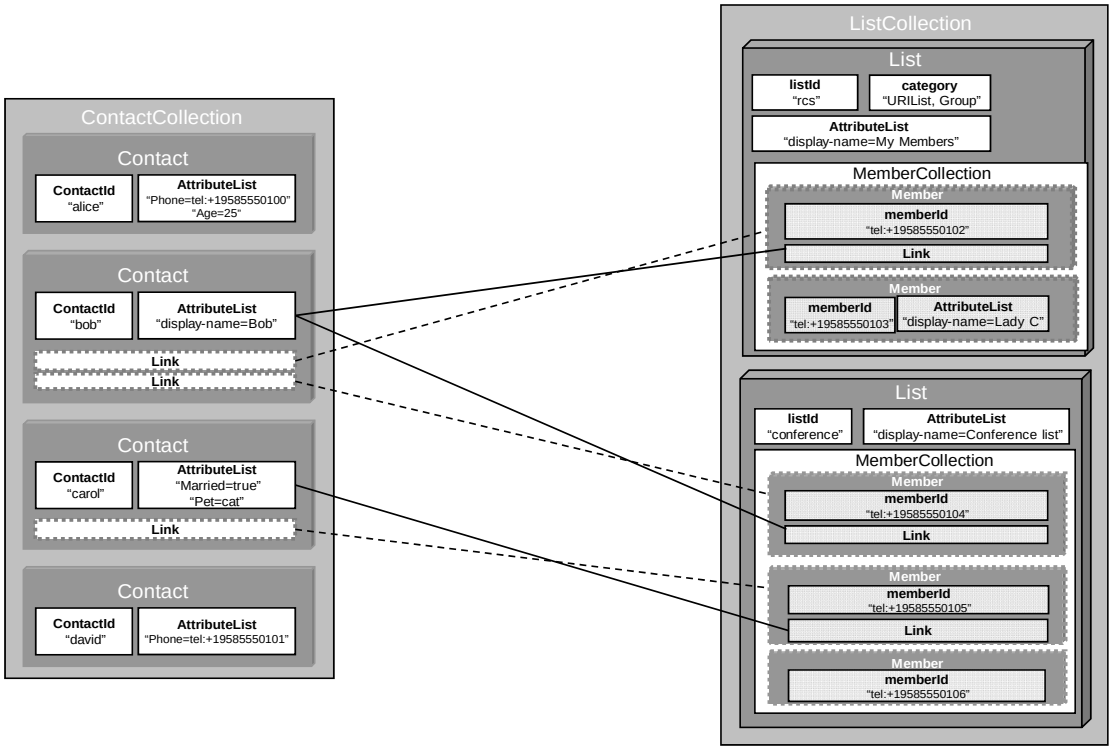

Figure 6 Data types for contacts, members and lists