



RESTful bindings for Parlay X Web Services – Address List Management

Candidate Version 1.0 – 11 Jan 2011

Open Mobile Alliance
OMA-TS-ParlayREST_AddressListManagement-V1_0-20110111-C

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

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

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

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

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

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

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.

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

Contents

1.	SCOPE	11
2.	REFERENCES	12
2.1	NORMATIVE REFERENCES	12
2.2	INFORMATIVE REFERENCES	12
3.	TERMINOLOGY AND CONVENTIONS	13
3.1	CONVENTIONS	13
3.2	DEFINITIONS	13
3.3	ABBREVIATIONS	13
4.	INTRODUCTION	14
4.1	VERSION 1.0	14
5.	ADDRESS LIST MANAGEMENT API DEFINITION	15
5.1	RESOURCES SUMMARY	15
5.2	ADDRESS LIST MANAGEMENT PARLAYREST API DATA STRUCTURES	22
5.2.1	Type: ContactCollection	22
5.2.2	Type: Contact.....	22
5.2.3	Type: SharedIdentity.....	23
5.2.4	Type: MemberListCollection	23
5.2.5	Type: MemberList	24
5.2.6	Type: AttributeList.....	24
5.2.7	Type: Attribute.....	25
5.2.8	Type: MemberCollection	25
5.2.9	Type: Member.....	25
5.2.10	Type: MemberListReferenceCollection	26
5.2.11	Type: ListChangesSubscriptionCollection	26
5.2.12	Type: ListChangesSubscription	27
5.2.13	Type: ListChangeNotification.....	27
5.2.14	Type: RuleList	28
5.2.15	Type: Rule.....	28
5.2.16	Type: ApplicableToContacts	29
5.2.17	Type: ApplicableToMemberLists	29
5.2.18	Type: ApplicableToMemberListsAndLinkedContacts	30
5.2.19	Type: AuthorizedTo.....	30
5.2.20	Enumeration: MemberListType	30
5.2.21	Enumeration: ResourceStatus	31
5.2.22	Values of the Link “rel” attribute.....	31
5.3	SEQUENCE DIAGRAMS	31
5.3.1	Managing contacts in a flat list	31
5.3.2	Accessing the member lists and members	33
5.3.3	Subscribing to list changes and receiving notifications	35
5.3.4	Managing shared member lists	36
5.3.5	Managing authorization rules.....	37
5.4	RESOURCE: COLLECTION OF CONTACTS	39
5.4.1	Request URI variables	40
5.4.2	Response Codes	40
5.4.2.1	<i>Response Codes</i>	40
5.4.2.2	<i>Exception fault codes</i>	41
5.4.3	GET.....	41
5.4.3.1	<i>Example 1: Retrieve all contacts with all attributes (default) (Informative)</i>	41
5.4.3.2	<i>Example 2: Retrieve all contacts with selected attributes (Informative)</i>	42
5.4.4	PUT.....	43
5.4.5	POST.....	43
5.4.6	DELETE	43
5.5	RESOURCE: INDIVIDUAL CONTACT	43

- 5.5.1 Request URI variables 43
- 5.5.2 Response Codes 44
 - 5.5.2.1 Response Codes 44
 - 5.5.2.2 Exception fault codes 44
- 5.5.3 GET 44
 - 5.5.3.1 Example 1: Retrieve individual contact with all attributes (default) (Informative)..... 44
 - 5.5.3.2 Example 2: Retrieve contact with selected attributes (Informative)..... 44
- 5.5.4 PUT 45
 - 5.5.4.1 Example 1: Create a new contact (Informative) 45
 - 5.5.4.2 Example 2: Update an existing contact with links to an existing member (Informative) 46
 - 5.5.4.3 Example 3: Update an existing contact with links to non existing member (Informative) 47
- 5.5.5 POST 48
- 5.5.6 DELETE 48
 - 5.5.6.1 Example: Delete a contact (Informative) 48
- 5.6 RESOURCE: ATTRIBUTES FOR A CONTACT..... 49**
- 5.6.1 Request URI variables 49
- 5.6.2 Response Codes 49
 - 5.6.2.1 Response Codes 49
 - 5.6.2.2 Exception fault codes 49
- 5.6.3 GET 49
 - 5.6.3.1 Example: Get all attributes of a contact (Informative) 49
- 5.6.4 PUT 50
- 5.6.5 POST 50
- 5.6.6 DELETE 50
- 5.7 RESOURCE: INDIVIDUAL ATTRIBUTE FOR A CONTACT 50**
- 5.7.1 Request URI variables 50
- 5.7.2 Response Codes 51
 - 5.7.2.1 Response Codes 51
 - 5.7.2.2 Exception fault codes 51
- 5.7.3 GET 51
 - 5.7.3.1 Example 1: Retrieve a contact’s attribute (Informative) 51
 - 5.7.3.2 Example 2: Retrieve a non existing attribute (Informative) 51
- 5.7.4 PUT 52
 - 5.7.4.1 Example: Create or update a contact’s attribute (Informative) 52
- 5.7.5 POST 52
- 5.7.6 DELETE 52
 - 5.7.6.1 Example: Delete a contact’s attribute (Informative) 53
- 5.8 RESOURCE: MEMBER LISTS..... 53**
- 5.8.1 Request URI variables 53
- 5.8.2 Response Codes 53
 - 5.8.2.1 Response Codes 53
 - 5.8.2.2 Exception fault codes 54
- 5.8.3 GET 54
 - 5.8.3.1 Example 1: Retrieve contact lists with all attributes and all members with all member attributes (default) (Informative) 54
 - 5.8.3.2 Example 2: Get all member lists belonging to a user without member list attributes and without members (Informative) 55
 - 5.8.3.3 Example 3: Retrieve specific member lists attributes and specific member attributes (Informative) 56
 - 5.8.3.4 Example 4: Retrieve member lists with specific member lists attributes and without members data (Informative) 57
 - 5.8.3.5 Example 5: Retrieve specific member lists attributes and list of members without attributes (Informative) 58
 - 5.8.3.6 Example 6: Retrieve member lists without attributes and members with all member attributes (Informative) 59
- 5.8.4 PUT 60
- 5.8.5 POST 60
- 5.8.6 DELETE 60
- 5.9 RESOURCE: INDIVIDUAL MEMBER LIST..... 61**
- 5.9.1 Request URI variables 61
- 5.9.2 Response Codes 61
 - 5.9.2.1 Response Codes 61
 - 5.9.2.2 Exception fault codes 61
- 5.9.3 GET 61

5.9.3.1	Example 1: Retrieve a member list with all attributes and all member attributes (default) (Informative)	61
5.9.3.2	Example 2: Retrieve a member list without list and member attributes (Informative)	62
5.9.3.3	Example 3: Retrieve a member list with all attributes and selected member attributes (Informative)	63
5.9.3.4	Example 4: Retrieve a non existing member list (Informative)	63
5.9.4	PUT	64
5.9.4.1	Example: Create a member list (Informative)	64
5.9.5	POST	65
5.9.6	DELETE	65
5.9.6.1	Example: Delete a member list (Informative)	65
5.10	RESOURCE: ATTRIBUTES FOR A MEMBER LIST	65
5.10.1	Request URI variables	65
5.10.2	Response Codes	65
5.10.2.1	Response Codes	65
5.10.2.2	Exception fault codes	66
5.10.3	GET	66
5.10.3.1	Example: Get all attributes for a member lists belonging to Bob (Informative)	66
5.10.4	PUT	66
5.10.5	POST	66
5.10.6	DELETE	66
5.11	RESOURCE: INDIVIDUAL ATTRIBUTE FOR A MEMBER LIST	67
5.11.1	Request URI variables	67
5.11.2	Response Codes	67
5.11.2.1	Response Codes	67
5.11.2.2	Exception fault codes	67
5.11.3	GET	67
5.11.3.1	Example 1: Retrieve an attribute value (Informative)	67
5.11.3.2	Example 2: Retrieve a non existing attribute (Informative)	68
5.11.4	PUT	68
5.11.4.1	Example: Create an attribute (Informative)	68
5.11.5	POST	69
5.11.6	DELETE	69
5.11.6.1	Example: Delete an attribute (Informative)	69
5.12	RESOURCE: MEMBERS IN A MEMBER LIST	69
5.12.1	Request URI variables	70
5.12.2	Response Codes	70
5.12.2.1	Response Codes	70
5.12.2.2	Exception fault codes	70
5.12.3	GET	70
5.12.3.1	Example 1: Get all members for a member list belonging to a user with all attributes (default) (Informative)	70
5.12.3.2	Example 2: Get all members for a member list belonging to a user, without member attributes (Informative)	71
5.12.4	PUT	72
5.12.5	POST	72
5.12.6	DELETE	72
5.13	RESOURCE: INDIVIDUAL MEMBER IN A MEMBER LIST	72
5.13.1	Request URI variables	72
5.13.2	Response Codes	73
5.13.2.1	Response Codes	73
5.13.2.2	Exception fault codes	73
5.13.3	GET	73
5.13.3.1	Example 1: Retrieve a member (Informative)	73
5.13.3.2	Example 2: Retrieve a member and selected attributes (Informative)	73
5.13.3.3	Example 3: Retrieve a non existing member (Informative)	74
5.13.4	PUT	74
5.13.4.1	Example 1: Create a member without a link to the Contact (Informative)	74
5.13.4.2	Example 2: Update member with relation to non-existing contact (Informative)	75
5.13.5	POST	76
5.13.6	DELETE	76
5.13.6.1	Example: Delete a member (Informative)	76
5.14	RESOURCE: ATTRIBUTES FOR A MEMBER IN A MEMBER LIST	76
5.14.1	Request URI variables	76

5.14.2	Response Codes	77
5.14.2.1	Response Codes	77
5.14.2.2	Exception fault codes	77
5.14.3	GET	77
5.14.3.1	Example: Get all attributes of a member of a member list (Informative)	77
5.14.4	PUT	78
5.14.5	POST	78
5.14.6	DELETE	78
5.15	RESOURCE: INDIVIDUAL ATTRIBUTE FOR A MEMBER IN A MEMBER LIST.....	78
5.15.1	Request URI variables	78
5.15.2	Response Codes	78
5.15.2.1	Response Codes	78
5.15.2.2	Exception fault codes	78
5.15.3	GET	79
5.15.3.1	Example 1: Retrieve a member's attribute (Informative)	79
5.15.3.2	Example 2: Retrieve a non existing attribute (Informative)	79
5.15.4	PUT	80
5.15.4.1	Example: Create a member's attribute (Informative)	80
5.15.5	POST	80
5.15.6	DELETE	80
5.15.6.1	Example: Delete an attribute (Informative)	80
5.16	RESOURCE: MEMBER LIST REFERENCES	81
5.16.1	Request URI variables	81
5.16.2	Response Codes	81
5.16.2.1	Response Codes	81
5.16.2.2	Exception fault codes	81
5.16.3	GET	81
5.16.3.1	Example: Get all member list references in a member lists (Informative)	81
5.16.4	PUT	82
5.16.5	POST	82
5.16.6	DELETE	82
5.17	RESOURCE: INDIVIDUAL MEMBER LIST REFERENCE IN A MEMBER LIST	82
5.17.1	Request URI variables	82
5.17.2	Response Codes	83
5.17.2.1	Response Codes	83
5.17.2.2	Exception fault codes	83
5.17.3	GET	83
5.17.3.1	Example 1: Retrieve a member list reference (Informative)	83
5.17.3.2	Example 2: Retrieve a non existing member list reference (Informative)	83
5.17.4	PUT	84
5.17.4.1	Example: Create a member list reference (Informative)	84
5.17.5	POST	84
5.17.6	DELETE	85
5.17.6.1	Example: Delete a member list reference (Informative)	85
5.18	RESOURCE: LIST CHANGES SUBSCRIPTIONS	85
5.18.1	Request URI variables	85
5.18.2	Response Codes	85
5.18.2.1	Response Codes	85
5.18.2.2	Exception fault codes	85
5.18.3	GET	85
5.18.3.1	Example: Get member list changes subscriptions (Informative)	86
5.18.4	PUT	86
5.18.5	POST	86
5.18.5.1	Example: Create new subscription for member list changes notification (Informative)	87
5.18.6	DELETE	87
5.19	RESOURCE: INDIVIDUAL MEMBER LIST CHANGES SUBSCRIPTION	88
5.19.1	Request URI variables	88
5.19.2	Response Codes	88
5.19.2.1	Response Codes	88
5.19.2.2	Exception fault codes	88

5.19.3	GET	88
5.19.3.1	Example: Get member list changes subscription (Informative)	88
5.19.4	PUT	89
5.19.4.1	Example: Update/Extend duration of subscription (Informative)	89
5.19.5	POST	90
5.19.6	DELETE	90
5.19.6.1	Example: Delete a member list changes subscription (Informative)	90
5.20	RESOURCE: CLIENT RESOURCE FOR MEMBER LIST CHANGES NOTIFICATIONS	90
5.20.1	Request URI variables	90
5.20.2	Response Codes	90
5.20.2.1	Response Codes	90
5.20.2.2	Exception fault codes	90
5.20.3	GET	91
5.20.4	PUT	91
5.20.5	POST	91
5.20.5.1	Example 1: Notification for subscription with sendFullMemberListContent=true (default) (Informative)	91
5.20.5.2	Example 2: Notification for contact change subscription (Informative)	91
5.20.5.3	Example 3: Notification for expired contact subscription (Informative)	92
5.20.6	DELETE	92
5.21	RESOURCE: COLLECTION OF CONTACTS SHARED BY OTHER USER	92
5.21.1	Request URI variables	93
5.21.2	Response Codes	93
5.21.2.1	Response Codes	93
5.21.2.2	Exception fault codes	93
5.21.3	GET	93
5.21.3.1	Example 1: Get contacts shared by a user identified by otherUserId (Informative)	93
5.21.3.2	Example 2: Get contact lists shared by a user identified by otherUserId without contact lists attributes and without member attributes (Informative)	94
5.21.4	PUT	95
5.21.5	POST	95
5.21.6	DELETE	95
5.22	RESOURCE: INDIVIDUAL SHARED CONTACT	95
5.22.1	Request URI variables	95
5.22.2	Response Codes	96
5.22.2.1	Response Codes	96
5.22.2.2	Exception fault codes	96
5.22.3	GET	96
5.22.3.1	Example 1: Retrieve information about individual shared contact (Informative)	96
5.22.3.2	Example 2: Retrieve information about individual shared contact (Informative)	97
5.22.3.3	Example 3: Retrieve information about individual shared contact, with selected contact attributes (default) (Informative)	98
5.22.4	PUT	98
5.22.5	POST	98
5.22.6	DELETE	98
5.23	RESOURCE: COLLECTION OF SHARED MEMBER LISTS	98
5.23.1	Request URI variables	99
5.23.2	Response Codes	99
5.23.2.1	Response Codes	99
5.23.2.2	Exception fault codes	99
5.23.3	GET	99
5.23.3.1	Example 1: Get member lists shared by a user identified by otherUserId with all list attributes and all member attributes (default) (Informative)	99
5.23.3.2	Example 2: Get member lists shared by a user identified by otherUserId without member lists attributes, but with selected member attributes (Informative)	101
5.23.4	PUT	102
5.23.5	POST	102
5.23.6	DELETE	102
5.24	RESOURCE: INDIVIDUAL SHARED MEMBER LIST	102
5.24.1	Request URI variables	102
5.24.2	Response Codes	103

5.24.2.1	Response Codes	103
5.24.2.2	Exception fault codes	103
5.24.3	GET	103
5.24.3.1	Example 1: Get individual member list shared by a user identified by otherUserId, with all member list attributes and all member attributes (default) (Informative)	103
5.24.3.2	Example 2: Get individual member list shared by a user identified by otherUserId, with all member list attributes and no member attributes (default) (Informative)	104
5.24.4	PUT	104
5.24.5	POST	105
5.24.6	DELETE	105
5.25	RESOURCE: INDIVIDUAL MEMBER INFORMATION FROM SHARED MEMBER LIST	105
5.25.1	Request URI variables	105
5.25.2	Response Codes	105
5.25.2.1	Response Codes	105
5.25.2.2	Exception fault codes	105
5.25.3	GET	106
5.25.3.1	Example 1: Get individual member information about user identified as memberId from the list shared by a user identified by otherUserId (default) (Informative)	106
5.25.3.2	Example 2: Get individual member information about user identified as memberId from the list shared by a user identified by otherUserId (default) (Informative)	106
5.25.4	PUT	107
5.25.5	POST	107
5.25.6	DELETE	107
5.26	RESOURCE: AUTHORIZATION RULES	107
5.26.1	Request URI variables	107
5.26.2	Response Codes	108
5.26.2.1	Response Codes	108
5.26.2.2	Exception fault codes	108
5.26.3	GET	108
5.26.3.1	Example: Get authorization rules (Informative)	108
5.26.4	PUT	109
5.26.5	POST	109
5.26.5.1	Example: Create new rule for memberList and referenced contacts (Informative)	109
5.26.6	DELETE	110
5.27	RESOURCE: INDIVIDUAL AUTHORIZATION RULE	110
5.27.1	Request URI variables	110
5.27.2	Response Codes	110
5.27.2.1	Response Codes	110
5.27.2.2	Exception fault codes	110
5.27.3	GET	111
5.27.3.1	Example: Get authorization rule (Informative)	111
5.27.4	PUT	111
5.27.4.1	Example: Update an existing authorization rule (Informative)	111
5.27.5	POST	112
5.27.6	DELETE	112
5.28	RESOURCE: INDIVIDUAL AUTHORIZATION RULE DATA	113
5.28.1	Request URI variables	113
5.28.2	Response Codes	113
5.28.2.1	Response Codes	113
5.28.2.2	Exception fault codes	113
5.28.3	GET	113
5.28.3.1	Example: Retrieve data from an authorization rule (Informative)	113
5.28.4	PUT	114
5.28.4.1	Example: Authorize user by updating an existing authorization rule (Informative)	114
5.28.5	POST	114
5.28.6	DELETE	115
5.28.6.1	Example: Delete an authorized identity (Informative)	115
6.	FAULT DEFINITIONS	116
6.1	SERVICE EXCEPTIONS	116

6.1.1	SVC0240: Key property changes not allowed	116
6.2	POLICY EXCEPTIONS	116
6.2.1	POL0214: Too many resources.....	116
APPENDIX A.	CHANGE HISTORY (INFORMATIVE).....	117
A.1	APPROVED VERSION HISTORY	117
A.2	DRAFT/CANDIDATE VERSION 1.0 HISTORY	117
APPENDIX B.	STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....	120
B.1	SCR FOR PARLAYREST.ALM SERVER.....	120
B.2	SCR FOR PARLAYREST.ALM.CONTACTLISTS SERVER.....	120
B.3	SCR FOR PARLAYREST.ALM.INDIVIDUALCONTACTLIST SERVER	120
B.3.1	SCR for ParlayREST.ALM.AttributesForAContact Server.....	121
B.3.2	SCR for ParlayREST.ALM.IndividualAttributeForAContact Server.....	121
B.3.3	SCR for ParlayREST.ALM.MemberLists Server.....	121
B.3.4	SCR for ParlayREST.ALM.IndividualMemberList Server.....	121
B.3.5	SCR for ParlayREST.ALM.AttributesForAMemberList Server.....	122
B.3.6	SCR for ParlayREST.ALM.IndividualAttributeForAMemberList Server.....	122
B.3.7	SCR for ParlayREST.ALM.MemberInAMemberList Server.....	122
B.3.8	SCR for ParlayREST.ALM.IndividualMemberInAMemberList Server.....	123
B.3.9	SCR for ParlayREST.ALM.AttributesForAMemberInAMemberList Server.....	123
B.3.10	SCR for ParlayREST.ALM.IndividualAttributeForAMemberInAMemberList Server.....	123
B.3.11	SCR for ParlayREST.ALM.MemberListReferences Server.....	124
B.3.12	SCR for ParlayREST.ALM.IndivMemberListReference Server.....	124
B.3.13	SCR for ParlayREST.ALM.List.Subscr Server.....	125
B.3.14	SCR for ParlayREST.ALM.Individual.Subscr Server.....	125
B.3.15	SCR for ParlayREST.ALM.Notif Server.....	125
B.3.16	SCR for ParlayREST.ALM.Shared.ContactCollection Server.....	126
B.3.17	SCR for ParlayREST.ALM.Individual.Shared.Contact Server.....	126
B.3.18	SCR for ParlayREST.ALM.Shared.Lists Server.....	126
B.3.19	SCR for ParlayREST.ALM.Individual.Shared.List Server.....	126
B.3.20	SCR for ParlayREST.ALM.Member.Shared.List Server.....	127
B.3.21	SCR for ParlayREST.ALM.Authorization.RuleList Server.....	127
B.3.22	SCR for ParlayREST.ALM.Indiv.AuthorizationRule Server.....	127
B.3.23	SCR for ParlayREST.ALM.Indiv.AuthorizationRuleData Server.....	128
APPENDIX C.	APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE)	129
C.1	CREATE A SUBSCRIPTION FOR MEMBER LIST CHANGES NOTIFICATIONS.....	129
C.1.1	Example (Informative).....	130
C.1.1.1	Request.....	130
C.1.1.2	Response.....	131
C.2	CREATE AN AUTHORIZATION RULE	131
C.2.1	Example (Informative).....	132
C.2.1.1	Request.....	133
C.2.1.2	Response.....	133
APPENDIX D.	JSON EXAMPLES (INFORMATIVE)	134
D.1	RETRIEVE ALL CONTACTS WITH ALL ATTRIBUTES (DEFAULT) (SECTION 5.4.3.1).....	134
D.2	RETRIEVE INDIVIDUAL CONTACT WITH ALL ATTRIBUTES (DEFAULT) (SECTION 5.5.3.1).....	135
D.3	GET ALL ATTRIBUTES OF A CONTACT (SECTION 5.6.3.1).....	136
D.4	RETRIEVE A CONTACT'S ATTRIBUTE (SECTION 5.7.3.1)	137
D.5	CREATE OR UPDATE A CONTACT'S ATTRIBUTE (SECTION 5.7.4.1)	137
D.6	RETRIEVE SPECIFIC MEMBER LISTS ATTRIBUTES AND SPECIFIC MEMBER ATTRIBUTES (SECTION 5.8.3.3).....	138
D.7	RETRIEVE A MEMBER LIST WITH ALL ATTRIBUTES AND ALL MEMBER ATTRIBUTES (DEFAULT) (SECTION 5.9.3.1)	139
D.8	RETRIEVE A NON EXISTING MEMBER LIST (SECTION 5.9.3.4).....	140
D.9	GET ALL ATTRIBUTES FOR A MEMBER LISTS BELONGING TO BOB (SECTION 5.10.3.1)	140
D.10	RETRIEVE AN ATTRIBUTE VALUE (SECTION 5.11.3.1)	141

D.11 CREATE AN ATTRIBUTE(SECTION 5.11.4.1).....141

D.12 GET ALL MEMBERS FOR A MEMBER LIST BELONGING TO A USER WITH ALL ATTRIBUTES (DEFAULT) (SECTION 5.12.3.1) 142

D.13 RETRIEVE A MEMBER AND SELECTED ATTRIBUTES (SECTION 5.13.3.2).....143

D.14 GET ALL ATTRIBUTES OF A MEMBER OF A MEMBER LIST (SECTION 5.14.3.1)143

D.15 RETRIEVE A NON EXISTING ATTRIBUTE (SECTION 5.15.3.2)144

D.16 CREATE A MEMBER’S ATTRIBUTE (SECTION 5.15.4.1).....144

D.17 DELETE AN ATTRIBUTE (SECTION 5.16.3.1).....145

D.18 CREATE A MEMBER LIST REFERENCE (SECTION 5.17.4.1)146

D.19 GET MEMBER LIST CHANGES SUBSCRIPTIONS (SECTION 5.18.3.1)146

D.20 GET MEMBER LIST CHANGES SUBSCRIPTION (SECTION 5.19.3.1)147

D.21 NOTIFICATION FOR CONTACT CHANGE SUBSCRIPTION (SECTION 5.20.5.2).....148

D.22 GET CONTACTS SHARED BY A USER IDENTIFIED BY OTHERUSERID (SECTION 5.21.3.1)149

D.23 CREATE A MEMBER WITHOUT A LINK TO THE MEMBER (SECTION 5.22.3.2).....150

D.24 GET MEMBER LISTS SHARED BY A USER IDENTIFIED BY OTHERUSERID WITHOUT MEMBER LISTS ATTRIBUTES, BUT WITH SELECTED MEMBER ATTRIBUTES (SECTION 5.23.3.2)150

D.25 CREATE A NEW MEMBER OF THE MEMBER LIST (SECTION 5.24.3.1)151

D.26 GET INDIVIDUAL MEMBER INFORMATION ABOUT USER IDENTIFIED AS MEMBERID FROM THE LIST SHARED BY A USER IDENTIFIED BY OTHERUSERID (SECTION 5.25.3.2)152

D.27 GET AUTHORIZATION RULES (SECTION 5.26.3.1).....153

D.28 UPDATE AN EXISTING AUTHORIZATION RULE (SECTION 5.27.4.1)154

D.29 RETRIEVE DATA FROM AN AUTHORIZATION RULE (SECTION 5.28.3.1).....155

D.30 AUTHORIZE USER BY UPDATING AN EXISTING AUTHORIZATION RULE (SECTION 5.28.4.1).....155

D.31 DELETE AN AUTHORIZED IDENTITY (SECTION 5.28.6.1).....156

APPENDIX E. PARLAY X OPERATIONS MAPPING (INFORMATIVE).....157

APPENDIX F. LIGHT-WEIGHT RESOURCES FOR ADDRESS LIST MANAGEMENT (INFORMATIVE) ..158

APPENDIX G. OVERVIEW OF DATA TYPES (INFORMATIVE).....159

Figures

Figure 1 Resource structure defined by this specification.....16

Figure 2 Managing contacts.....32

Figure 3 Accessing member lists and members.....34

Figure 4 Subscription to notifications about list changes.....35

Figure 5 Flow for managing shared member lists.....37

Figure 6 Data types for contacts, members and member lists159

Tables

Table 1: listFilter query parameter definition.....39

Table 2: indivFilter query parameter definition40

Table 3: Parlay X operations mapping157

Table 4: Light-weight resources for Address List Management158

1. Scope

This specification defines a RESTful Address List Management 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/>
- [OMA_REST_TS_Common] “Common definitions and specifications for OMA REST interfaces”, Open Mobile Alliance™, OMA-TS_REST_Common-V1_0, URL: <http://www.openmobilealliance.org/>
- [REST_TS_Common] “RESTful bindings for Parlay X Web Services – Common”, Open Mobile Alliance™, OMA-TS-ParlayREST_Common-V1_1, URL:<http://www.openmobilealliance.org/>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL:<http://www.ietf.org/rfc/rfc2616.txt>
- [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] W3C HTML 2.0 Specification, form-urlencoded Media Type, URL: http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1
- [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/>
- [REST_WP] “White Paper on Guidelines for REST API specifications”, Open Mobile Alliance™, OMA-WP-Guidelines_for_REST_API_specifications, 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 [OMA-DICT].

3.3 Abbreviations

ALM	Address List Management
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
TS	Technical Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The ParlayREST Technical Specification for Address List Management (ALM) contains the HTTP protocol binding for the Parlay X Address List Management Web Services 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 form-urlencoded).

Generic guidelines for REST API specification development in OMA can be found in [REST_WP].

4.1 Version 1.0

Version 1.0 of Address List Management ParlayREST API specification supports the following operations:

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

5. Address List Management API definition

This section is organized to support a comprehensive understanding of the Address List Management 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_TS_Common] and [OMA_REST_TS_Common].

The remainder of this document is structured as follows:

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

The remaining subsections in section 5 contain the detailed specification for each of the resources. Each such subsection defines the resource, the request URI 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 5 use XML as the format for the message body. Form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D. Appendix B provides the Static Conformance Requirements (SCR). Appendix E lists the Parlay X equivalent method for each supported ParlayREST resource and method combination, where applicable. Appendix F lists all Address List Management data structure elements that can be accessed individually as light-weight resources. Finally, Appendix G illustrates the most common data structures and their relationships in Address List Management.

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 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 [OMA_REST_TS_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.

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

5.1 Resources Summary

This section summarizes all the resources used by the Address List Management API.

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.

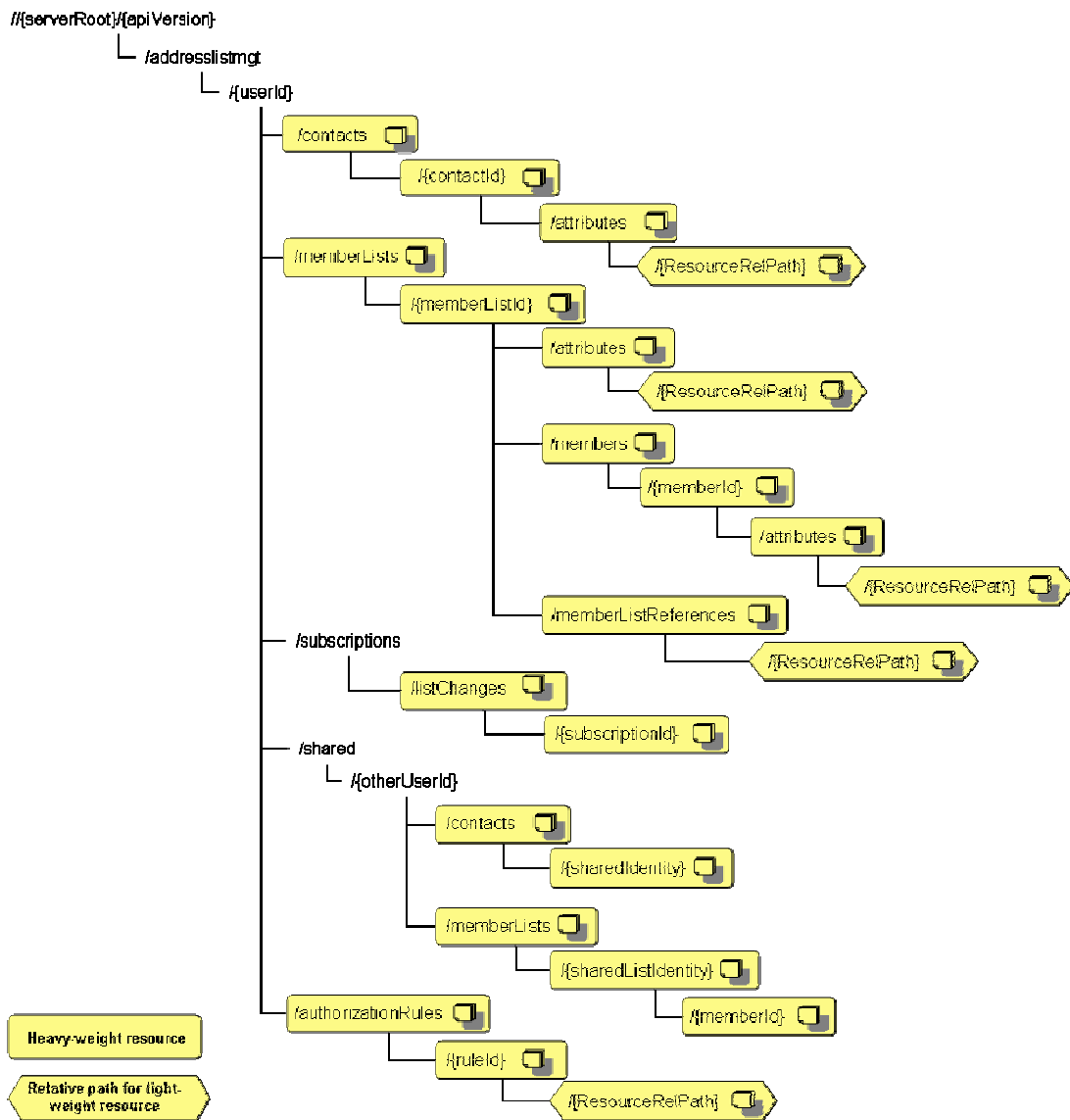


Figure 1 Resource structure defined by this specification

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}/{apiVersion}/addresslistmgt	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	This operation updates the entire list of attributes.	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 member lists

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Member lists	/ {userId}/memberLists	MemberListCollection	This operation retrieves all member lists the user has created.	No	No	No
Individual member list	/ {userId}/memberLists/{memberListId}	MemberList	This operation retrieves the member list.	This operation creates or updates the entire member list.	No	This operation removes the member list from the system including

							attributes and members
Attributes for a member list		/ {userId}/memberLists/{memberListId}/attributes	AttributeList	This operation returns all attributes for a member list	This operation updates the entire list of attributes.	No	No
Individual attribute for a member list		/ {userId}/memberLists/{memberListId}/attributes/[ResourceRelPath]	Attribute	This operation returns the value of the member list attribute	This operation creates or updates an attribute	No	This operation deletes an attribute
Members in a member list		/ {userId}/memberLists/{memberListId}/members	MemberCollection	This operation retrieves the members in the member list.	No	No	No
Individual member in a member list		/ {userId}/memberLists/{memberListId}/members/{memberId}	Member	This operation retrieves a user from a member list. (Normally only used to verify the existence of the member)	This operation creates and updates a member of the member list.	No	This operation removes the member from the member list.
Attributes for a member in a member list		/ {userId}/memberLists/{memberListId}/members/{memberId}/attributes	AttributeList	This operation returns all attributes for a member MemberDefiner	This operation updates the entire list of attributes.	No	No
Individual attribute for a member in a member list		/ {userId}/memberLists/{memberListId}/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 member	No	This operation removes an attribute for a member
Member list references		/ {userId}/memberLists/{memberListId}/memberListReferences	MemberListReferenceCollection	This operation returns a list of references to other member lists.	No	No	No

Individual member list reference		/{userId}/memberLists/{memberListId}/memberListReferences/[ResourceRelPath]	Link	This operation returns the value of member list reference	This operation creates a new member list reference.	No	This operation removes a member list reference
----------------------------------	--	---	------	---	---	----	--

Purpose: To allow client to manage subscriptions to list changes

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
List changes subscriptions	/{userId}/subscriptions/listChanges	ListChangesSubscriptionCollection	This operation retrieves all active subscriptions for list changes	No	Creates new subscription for list changes	No
Individual list changes subscription	/{userId}/subscriptions/listChanges/{subscriptionId}	ListChangesSubscription	This operation retrieves information about individual subscription	Updates/extends the duration of the subscription	No	This operation deletes and terminates subscription

Purpose: To allow client to receive notifications about contact list changes

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Client resource for list changes notifications	<specified by client in the subscription>	ListChangeNotification	No	No	Notification about list change	No

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

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	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 requesting user identified by {userId}	No	No	No
Individual shared contact	/ {userId}/shared/{otherUserId}/contacts/{sharedIdentity}	Contact	This operation retrieves information about individual member list shared by user identified by {otherUserId} with requesting user identified by {userId}	No	No	No

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

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of shared member lists	/ {userId}/shared/{otherUserId}/memberLists	MemberListCollection	This operation retrieves all member lists shared by another user identified by {otherUserId} with the requesting user identified by {userId}	No	No	No

Individual shared member list	/ {userId}/shared/{otherUserId}/memberLists/{sharedListIdentity}	MemberList	This operation retrieves information about individual member list shared by user identified by {otherUserId} with requesting user identified by {userId}	No	No	No
Individual member information from shared member list	/ {userId}/shared/memberLists/{otherUserId}/{sharedListIdentity}/{memberId}	Member	This operation retrieves individual member information from the shared member list	No	No	No

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

Resource	Base URL: http://{serverRoot}/{apiVersion}/addresslistmgt	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)	Retrieve all authorization rules	No	Create new authorization rule	No
Individual authorization rule	/ {userId}/authorizationRules/{ruleId}	Rule (Used for PUT/GET)	Retrieves individual authorization rule	Updates individual authorization rule	No	Removes individual authorization rule
Individual authorization rule data	/ {userId}/authorizationRules/{ruleId}/[ResourceRelPath]	<i>The data structure corresponds to the element pointed out by the request-URI.</i> (Used for PUT/GET)	Retrieves data as specified in the URI	Creates or updates data as specified in the URI	No	Removed the data as specified in the URI

5.2 Address List Management ParlayREST API Data Structures

The namespace for the Address List Management data types is:

urn:oma:xml:rest:addresslistmgt: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 [OMA_REST_TS_Common]. The use of the names 'xsd' and 'common' is not semantically significant.

5.2.1 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	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.2 Type: Contact

Individual contact information in the flat unique list of contacts

Element/Attribute	Type	Optional	Description
contactId	xsd:anyURI	Yes	Contains a relative URI of the contact (e.g. 'Alice'). The element is mandatory when used inside 'ContactCollection' type. It is a key property of the element. It is defined as an attribute in XML format
sharedIdentity	SharedIdentity	Yes	Contains a list of publicly known identities (absolute URI:s) of the contact (e.g. mailto:alice@example.com, tel:+1-555-555-0100 etc). 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: SHALL NOT be included in POST requests, MUST 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 MemberList associated with the contact. When a member containing a link to a contact is added to a member 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 member 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 member list shared with the requester.

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

5.2.3 Type: SharedIdentity

Contains a list of publicly known identities.

Element	Type	Optional	Description
sharedId	xsd:anyURI [0..unbounded]	Yes	Contains a list of absolute URI's representing identities known to other users, used when retrieving a Contact or Member list.

5.2.4 Type: MemberListCollection

Contains a list of member lists.

Element	Type	Optional	Description
memberList	MemberList [0..unbounded]	Yes	Contains a list of member list identities.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named memberListCollection of type MemberListCollection is allowed in response bodies.

5.2.5 Type: MemberList

Contains a list of identities.

Element/Attribute	Type	Optional	Description
memberListId	xsd:anyURI	Yes	Contains a relative URI representing the member list identity (e.g. "friends"). The element is mandatory when used inside the 'MemberListCollection' type. It is a key property of the element. It is defined as an attribute in XML format
memberCollection	MemberCollection	Yes	Members in the member list
memberListReferenceCollection	MemberListReferenceCollection	Yes	Contains references to other member lists.
category	MemberListType [0..unbounded]	Yes	Specifies the type of member list. URIList is default if not specified.
sharedListIdentity	SharedIdentity	Yes	Contains a list of publicly known identities (absolute URI:s) of the member list (e.g. mailto:alice@example.com;list=friends, tel:+1-555-555-0100;list=friends etc). A shared identity MAY be used by a requesting user to identify the member list in addition to the memberListId (since the memberListId may not be known by others).
attributeList	AttributeList	Yes	Contains a list of attributes related to a contacts list.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.6 Type: AttributeList

Attributes of a member list

Element	Type	Optional	[ResourceRelPath]	Description
attribute	Attribute [0..unbounded]	Yes	{name}	Contains a list of attributes related to contact, member 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	Yes	Not applicable	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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].

5.2.7 Type: Attribute

Individual attribute of a member list

Element/Attribute	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. It is defined as an attribute in XML format.
value	xsd:anyType	No	Value of the attribute. The base type for this element to be used is the xsd:string type but also other types can be used when appropriate (e.g. an attribute that is a URI can be transported using an xsd:anyURI type). It is also possible to use more complex types if an attribute cannot be described in a simple way using a string or if it already exist a defined complex type that can be reused. The type definition for a particular attribute is left out of scope of this specification.

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

5.2.8 Type: MemberCollection

Collection of members that are part of a member list

Element	Type	Optional	Description
member	Member [0..unbounded]	Yes	Contains a list of members related to a members list.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.9 Type: Member

Individual member information in a member list

Element	Type	Optional	Description
memberId	xsd:anyURI	Yes	Contains an identifier of a single member (e.g. tel URI). The element is mandatory when used inside 'MemberCollection' type. It is a key property of the element. It is defined as an attribute in XML format
attributeList	AttributeList	Yes	Contains a list of attributes (e.g. display name) related to a member.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST 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 a (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 (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 member 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.

5.2.10 Type: MemberListReferenceCollection

Contains a list of references to other member lists.

Element	Type	Optional	[ResourceRelPath]	Description
resourceURL	xsd:anyURI	Yes	Not applicable	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link [0..unbounded]	Yes	{href}	Contains references to other member lists (rel="MemberList").

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

5.2.11 Type: ListChangesSubscriptionCollection

Collection of subscriptions to notifications of changes in contact lists

Element	Type	Optional	Description
listChangesSubscription	ListChangesSubscription [0..unbounded]	Yes	Contains a list of subscriptions related to list changes.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named `contactListChangesSubscriptionCollection` of type `ContactListChangesSubscriptionCollection` is allowed in response bodies.

5.2.12 Type: ListChangesSubscription

Individual subscription to notifications of changes in a contact list

Element	Type	Optional	Description
<code>anyContacts</code>	empty	Choice	Indicates a subscription for changes about any contacts in the <code>ContactCollection</code> .
<code>memberListId</code>	<code>xsd:anyURI</code>	Choice	Indicates a subscription for changes about the specified member list.
<code>callbackReference</code>	<code>common:CallbackReference</code>	No	Client's Notification URL and OPTIONAL <code>callbackData</code>
<code>clientCorrelator</code>	<code>xsd:string</code>	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.
<code>applicationTag</code>	<code>xsd:string</code>	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.
<code>duration</code>	<code>xsd:int</code>	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.
<code>resourceURL</code>	<code>xsd:anyURI</code>	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

A root element named `listChangesSubscription` of type `ListChangesSubscription` is allowed in request and response bodies.

5.2.13 Type: ListChangeNotification

A notification SHALL be generated in the following cases only:

- member or contact is added to a list
- member or contact is removed from a list
- attribute related to a member or contact is updated (attribute added, attribute value updated or attribute removed)
- attribute related to a member list is updated (attribute added, attribute value updated or attribute removed)
- to indicate that the subscription has been terminated by the server.

(ResourceState=TerminatedTimeout/TerminatedNoResource/TerminatedOther)

Element	Type	Optional	Description
callbackData	xsd:string	Yes	CallbackData if passed by the application during the associated subscription.
resourceStatus	ResourceStatus	No	Indicates the state for the subscription.
duration	xsd:int	Yes	Subscription duration in seconds. Server would expire the subscription after specified number of seconds after subscription creation.
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="MemberList" or rel="ContactCollection" respectively). SHALL contain reference to the subscription (rel="ListChangesSubscription") if resourceStatus is 'Active'.

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

5.2.14 Type: RuleList

Contains a list of authorization rules.

Element	Type	Optional	Description
rule	Rule [0..unbounded]	Yes	Contains a list of all authorization rules related to contacts and member lists.
resourceURL	xsd:anyURI	Yes	Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

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

5.2.15 Type: Rule

Defines an individual authorization rule.

Element/Attribute	Type	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 lightweight resource URL then it SHALL NOT be altered. It is defined as an attribute in XML format
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	ApplicableToContacts	Yes	Defines for which contacts the rule applies for.
applToMemberLists	ApplicableToMemberLists	Yes	Defines for which member lists the rule applies for.
applToMemberListsAndContacts	ApplicableToMemberListsAndLinkedContacts	Yes	Defines for which member lists and referenced contacts the rule applies for.
authorizedTo	AuthorizedTo	No	Defines who have access to member lists and contacts.
listFilter	xsd:string [0..unbounded]	Yes	Filter for member list attributes (according to 5.4) 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 (according to 5.4) 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. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.

5.2.16 Type: ApplicableToContacts

Specifies what contacts the rule applies to.

Element/Attribute	Type	Optional	Description
contactId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified contact identity in the ContactCollection.
allContacts	(empty)	Choice	The rule applies to all contacts.

5.2.17 Type: ApplicableToMemberLists

Specifies what member lists the rule applies to.

Element/Attribute	Type	Optional	Description
memberListId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified member list.
allMemberLists	(empty)	Choice	The rule applies to all member lists.

5.2.18 Type: ApplicableToMemberListsAndLinkedContacts

Specifies what member lists and referenced contacts the rule applies to.

Element/Attribute	Type	Optional	Description
memberListId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified member list. The rule applies also to contacts that are referenced from the specified member list.
allMemberLists	(empty)	Choice	The rule applies for all member lists and linked contacts.

5.2.19 Type: AuthorizedTo

Specifies authorized identities.

Element/Attribute	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	<i>Not applicable</i>	All contacts with shared identities are authorized to access a resource for which the rule is applicable to.
memberListId	xsd:anyURI [0..unbounded]	Choice	memberLists/{memberListId}	Users within the member list are authorized to access a resource for which the rule is applicable to.
allMemberLists	(empty)	Choice	<i>Not applicable</i>	Users within all member lists are authorized to access a resource for which the rule is applicable to.
link	common:Link [0..unbounded]	Choice	externalMemberLists/{href}	Users within the member list (owned by another user) are authorized to access a resource for which the rule is applicable to. Link specified using rel="MemberList".
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	<i>Not applicable</i>	Any user is authorized to access a resource for which the rule is applicable to.

5.2.20 Enumeration: MemberListType

Enumerates types of member lists.

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

5.2.21 Enumeration: ResourceStatus

Enumerates the 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.
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.

5.2.22 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
- ListChangesSubscription
- ListChangesSubscriptionCollection
- Member
- MemberCollection
- MemberList
- MemberListCollection
- MemberListReferenceCollection

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

5.3 Sequence Diagrams

5.3.1 Managing contacts in a flat list

The figure below shows various ways to retrieve and manage contacts in a flat list. 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 flat list of contacts.

The resources:

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

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

- To retrieve all attributes of a specific contact the following resource is used:

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

- To retrieve, create, update or delete a single attribute of a specific contact the following resource is used:

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

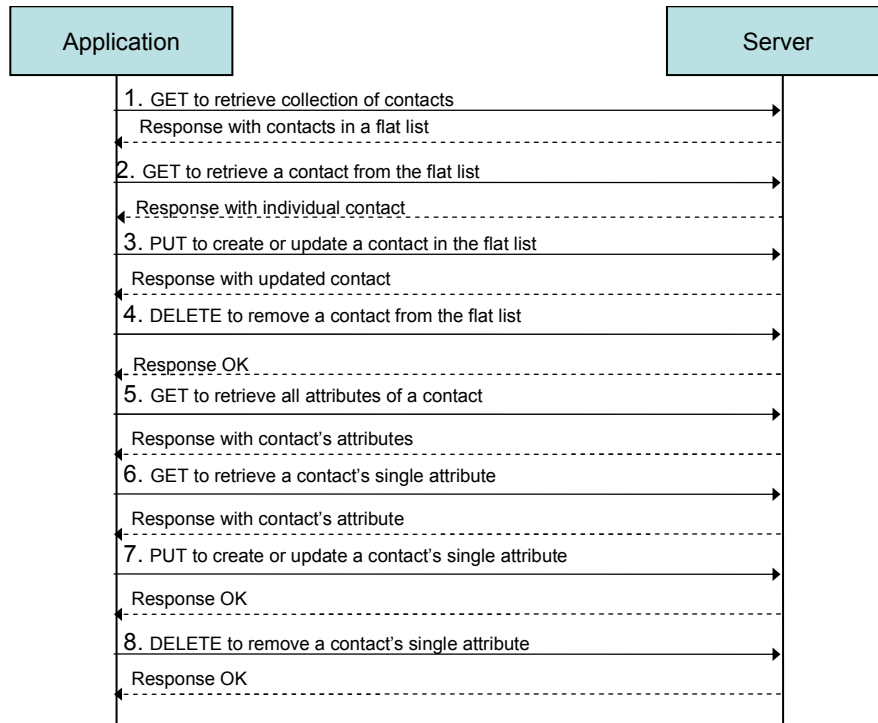


Figure 2 Managing contacts

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

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

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

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

5.3.2 Accessing the member lists and members

The figure below shows various ways to retrieve and manipulate data in member lists. There is one application acting on behalf of Alice. The application is interested in one specific member 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 member lists the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists
- To add of a member list the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}
- To fetch all the member s of a list the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members
- To add a specific member to a member list the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}
- To fetch all attributes for a member list this resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/attributes
- To fetch all attributes of a specific member the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}/attributes
- To fetch a single attribute of a specific member the following resource is used:
http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}/attributes/[ResourceRelPath]

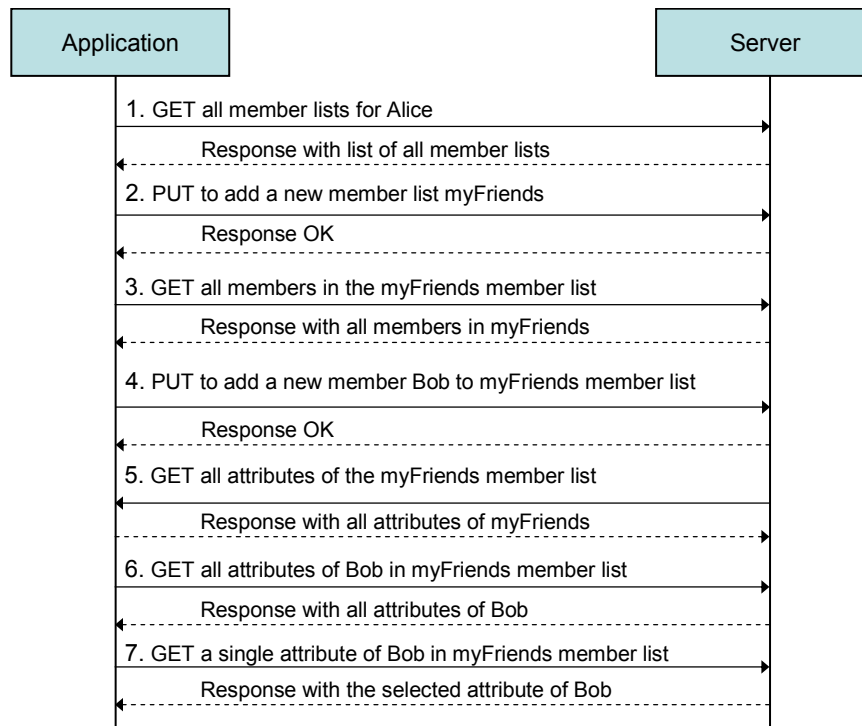


Figure 3 Accessing member lists and members

1. The application fetches all the lists for the userId Alice by doing a GET on the following resource:
http://{serverRoot}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists
 The result contains all the member lists for Alice.
2. The application adds a new member list called myFriends by doing a PUT on the following resource:
http://{serverRoot}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/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}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/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}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/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}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/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}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/myFriends/members/

{bobUserId}/attributes

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

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

The result contains Bob's display name attribute.

5.3.3 Subscribing to list changes and receiving notifications

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

The resources:

- To retrieve all the subscriptions to member lists changes, and to create a new subscription the following resource is used:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/subscriptions/listChanges

- To retrieve or delete an individual subscription, the following resource is used:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/subscriptions/listChanges/{subscriptionId}

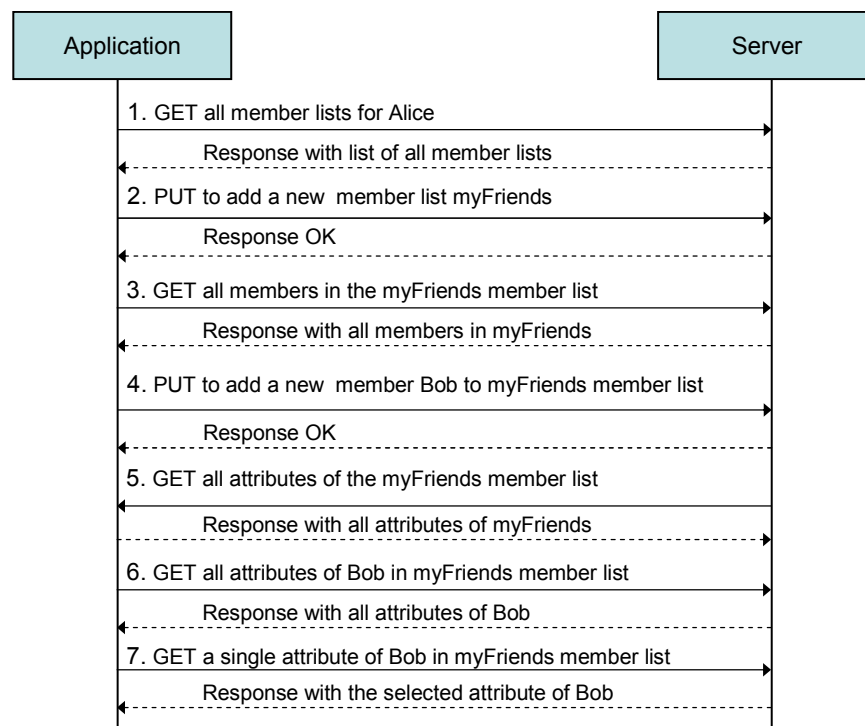


Figure 4 Subscription to notifications about list changes

- The application retrieves list of all subscriptions for member list changes for the userId by performing a GET on the following resource:

http://{serverRoot}/{apiVersion}/addresslistmgt/{aliceUserId}/subscriptions/listChanges

The result contains all the subscriptions created by Alice.

- The application retrieves an individual subscription identified by subscriptionId by performing a GET on the following resource:

http://{serverRoot}/{apiVersion}/addresslistmgt/{aliceUserId}/subscriptions/listChanges/{subscriptionId}

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

3. The application creates a new subscription for list changes notification by performing a POST on the following resource, including in the request body a client application supplied notifyURL:

http://{serverRoot}/{apiVersion}/addresslistmgt/{aliceUserId}/subscriptions/listChanges

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}/{apiVersion}/addresslistmgt/{aliceUserId}/memberLists/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 member 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}/{apiVersion}/addresslistmgt/{aliceUserId} /subscription/listChanges/{subscriptionId}

The result was successful.

5.3.4 Managing shared member lists

The figure below shows how to manage shared member lists.

The resources:

- To retrieve all member lists shared by another user the following resource is used:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists

- To retrieve an individual member list shared by another user the following resource is used:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists/{sharedListIdentity}
}

- To retrieve a member's information from a member list shared by another user the following resource is used:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists/{sharedListIdentity}
}/members/{memberId}

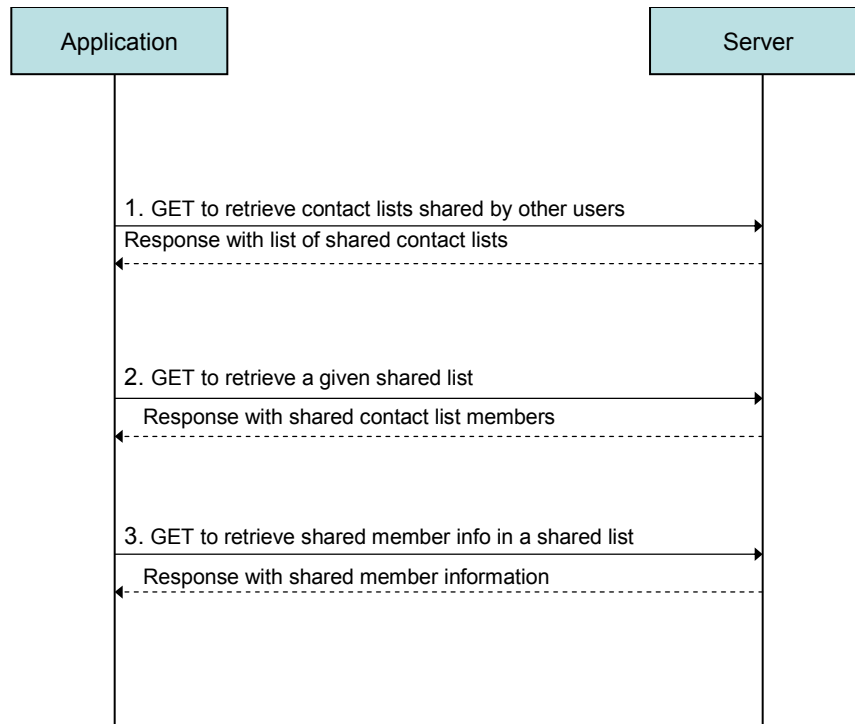


Figure 5 Flow for managing shared member lists

1. The application retrieves a list of all member lists shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists`

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

2. The application retrieves an individual member list shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists/{sharedListIdentity}`

The result contains all the members in the shared member list.

3. The application retrieves member's information from a member list shared by the user identified by otherUserId with the user identified by userId, by performing a GET on the following resource:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/memberLists/{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 member lists.

The resources:

- To retrieve all authorization rules and to create a new authorization rule the following resource is used:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/authorizationRules`

- To retrieve, update or delete an individual authorization rule the following resource is used:

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

- To retrieve, update or delete the authorized entity in an authorization rule the following resource is used:

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

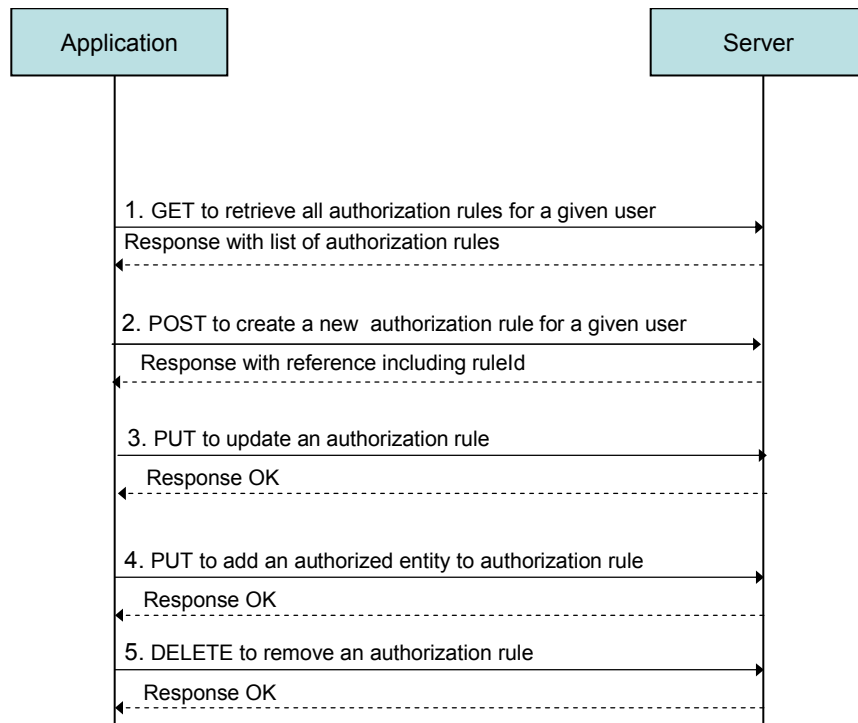


Figure 6 Flow for managing authorization rules

1. The application retrieves a list of all authorization rules for a given user/member lists shared by the user identified by userId, by performing a GET on the following resource:

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

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

5.4 Resource: Collection of contacts

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/contacts?indivFilter={indivFilterValue}

The `userId` must be percent-encoded according to [RFC3986].

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

The following OPTIONAL feature is introduced to allow the client to control the amount of information returned in the response body to a GET method executed on Address List Management resources. This feature SHALL NOT be supported for POST, PUT and DELETE methods executed on Address List Management resources.

This feature specifies the use of `listFilter` as a query parameter that controls the information returned relative to the list, and the use of `indivFilter` as a query parameter that controls the information returned relative to the members/contacts in the list/collection. These query parameters can be used in conjunction or independently. These query parameters SHALL not be in resource URLs that are part of resource representations.

In case the feature is NOT supported, the Server SHALL ignore the query parameters `listFilter` and `indivFilter`, and SHALL return the default amount of information in GET response body.

In case the feature is supported, the table below specifies the normative use of the query parameter `listFilter`:

Query parameter <code>listFilter</code>	Query parameter <code>listFilterValue</code>	Description
Not applicable (<code>listFilter</code> absent)	Not applicable	GET response body SHALL include all the list sub-elements, plus all list attributes (default behaviour).
<code>listFilter=</code>	List attribute name (e.g. <code>name1</code> , <code>name2</code> , ...).	Any number of <code>listFilter=listFilterValue</code> pairs separated by & are allowed. GET response body SHALL include all the list sub-elements, plus only selected list attributes (i.e. <code>name1</code> , <code>name2</code> , ...).
<code>listFilter=</code>	<code>~noAttr</code>	GET response body SHALL include all the list sub-elements and none of the list attributes.

Table 1: `listFilter` query parameter definition

In case the feature is supported, the table below specifies the normative use of the query parameter `indivFilter`:

Query parameter <code>indivFilter</code>	Query parameter <code>indivFilterValue</code>	Description
Not applicable (<code>indivFilter</code> absent)	Not applicable	GET response body SHALL include all member/contact sub-elements,

		plus all member/contact attributes (default behaviour).
indivFilter=	Member or Contact attribute name (e.g. name1, name2, ...).	Any number of indivFilter=indivFilterValue pairs separated by & are allowed. GET response body SHALL include all member/contact sub-elements, plus only selected member/contact attributes (i.e. name1, name2, ...).
indivFilter=	~noAttr	GET response body SHALL include all member/contact sub-elements and none of the member/contact attributes. This value SHALL only be supported for GET method on a list, and SHALL NOT be supported on GET method on member/contact in a list/collection.
indivFilter=	~none	GET response body SHALL include none of the member/contact information (member/contact structure to be omitted). This value SHALL only be supported for GET method on a list, and SHALL NOT be supported on GET method on member/contact in a list/collection or on a GET method on the contact collection.

Table 2: indivFilter query parameter definition

5.4.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
indivFilter	defines the level of information related to contacts of the collection, returned in a GET response (see 5.4 for details). indivFilter=~none is not supported for this resource.

5.4.2 Response Codes

5.4.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.4.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.4.3 GET

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

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

5.4.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/contacts HTTP/1.1
Accept: application/xml
Host: example.com
```

5.4.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contactCollection xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1">
  <contact contactId="alice">
    <attributeList>
      <attribute name="display-name">
        <value>Alice</value>
      </attribute>
      <attribute name="cellphone">
        <value>tel:+1-555-555-0200</value>
      </attribute>
      <attribute name="state">
        <value>California</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/contacts/alice</resourceURL>
    <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
  </contact>
  <contact contactId="sam">
    <sharedIdentity>
      <sharedId>tel:+1-555-555-0121</sharedId>
    </sharedIdentity>
    <attributeList>
      <attribute name="display-name">
        <value>Sam</value>
      </attribute>
      <attribute name="cellphone">
        <value>tel:+1-555-555-0300</value>
      </attribute>
      <attribute name="state">
        <value>New Jersey</value>
      </attribute>
    </attributeList>
  </contact>
</alm:contactCollection>
```

```

</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/sam</resourceURL>
<link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0121"/>
<link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2B1-555-555-0121"/>
</contact>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts</resourceURL>
</alm:contactCollection>

```

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

5.4.3.2.1 Request

```

GET ../{apiVersion}/addresslistmgt/tel%3A%2b1-555-555-0100/contacts?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com

```

5.4.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contactCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <contact contactId="alice">
    <attributeList>
      <attribute name="cellphone">
        <value>tel:+1-555-555-0200</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2b1-555-555-0100/contacts/alice</resourceURL>
    <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
  </contact>
  <contact contactId="sam">
    <sharedIdentity>
      <sharedId>tel:+1-555-555-0121</sharedId>
    </sharedIdentity>
    <attributeList>
      <attribute name="cellphone">
        <value>tel:+1-555-555-0300</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/sam</resourceURL>
    <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0121"/>
    <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2B1-555-555-0121"/>
  </contact>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts</resourceURL>

```

```
</alm:contactCollection>
```

5.4.4 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].

5.4.5 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].

5.4.6 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].

5.5 Resource: Individual contact

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/contacts/{contactId}?indivFilter={indivFilterValue}

The userId and contactId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of indivFilter query parameters.

5.5.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
contactId	contact identifier. Example: alice
indivFilter	defines the level of information related to contacts of the collection, returned in a GET response (see 5.4 for details). indivFilter=~none and indivFilter=~noAttr are not supported for this resource.

5.5.2 Response Codes

5.5.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.5.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.5.3 GET

This resource is used to retrieve individual member data from unique list of members from all member lists of a given user.

5.5.3.1 Example 1: Retrieve individual contact with all attributes (default) (Informative)

5.5.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice HTTP/1.1
Accept: application/xml
Host: example.com
```

5.5.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="alice">
  <attributeList>
    <attribute name="display-name">
      <value>Alice</value>
    </attribute>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0200</value>
    </attribute>
    <attribute name="state">
      <value>California</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
</alm:contact>
```

5.5.3.2 Example 2: Retrieve contact with selected attributes (Informative)

5.5.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

5.5.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="alice">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0200</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
</alm:contact>

```

5.5.4 PUT

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

5.5.4.1 Example 1: Create a new contact

(Informative)

5.5.4.1.1 Request

```

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="maria">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0250</value>
    </attribute>
  </attributeList>
</alm:contact>

```

5.5.4.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria
Date: Fri, 10 Dec 2010 12:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="maria">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0255</value>
    </attribute>
  </attributeList>
</alm:contact>

```

```

</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria</resourceURL>
</alm:contact>

```

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

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

5.5.4.2.1 Request

```

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="maria">
  <attributeList>
<attribute name="cellphone">
  <value>tel:+1-555-555-0255</value>
  </attribute>
</attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2b1-555-555-0255"/>
</alm:contact>

```

5.5.4.2.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Location:http://example.com/1/ addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria
Date: Fri, 10 Dec 2010 12:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="maria">
  <attributeList>
<attribute name="cellphone">
  <value>tel:+1-555-555-0255</value>
  </attribute>
</attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2B1-555-555-0255"/>
</alm:contact>

```

5.5.4.2.3 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/memberLists/5678 HTTP/1.1
Accept: application/xml
Host: example.com

```

5.5.4.2.4 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberListId="5678">
  <memberCollection>
    <member memberId="+1-555-555-0255">
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3AAliza@example.com
    </resourceURL>
    <link rel="Contact" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria"/>
    </member>
    <member memberId="mailto:liza@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Wife</value>
        </attribute>
        <attribute name="age">
          <value>42</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3AAliza@example.com
    </resourceURL>
    </member>
    <member memberId="mailto:serezha@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Son</value>
        </attribute>
        <attribute name="phone">
          <value>tel:+1-555-555-0260</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/members/mailto%3Aserezha@example.com
    </resourceURL>
    </member>
  </memberCollection>
  <attributeList>
    <attribute name="display-name">
      <value>Family</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</alm:memberList>

```

5.5.4.3 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 exist.

5.5.4.3.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria HTTP/1.1
```

```
Accept: application/xml
```

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

```
Host: example.com
```

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="maria">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0255</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2B1-555-555-0300"/>
</alm:contact>
```

5.5.4.3.2 Response

```
HTTP/1.1 403 Forbidden
```

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

```
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

5.5.5 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].

5.5.6 DELETE

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

5.5.6.1 Example: Delete a contact

(Informative)

5.5.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/contacts/sam HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com
```

5.5.6.1.2 Response

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

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


5.6 Resource: Attributes for a contact

The resource used is:

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

The userId and contactId must be percent-encoded according to [RFC3986].

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

5.6.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
contactId	contact identifier. Example: alice

5.6.2 Response Codes

5.6.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.6.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.6.3 GET

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

5.6.3.1 Example: Get all attributes of a contact (Informative)

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

5.6.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria/attributes?resFormat=XML HTTP/1.1
Host: example.com
```

5.6.3.1.2 Response

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

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:attributeList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
<attribute name="cellphone">
  <value>tel:+1-555-555-0255</value>
</attribute>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria/attributes</resourceURL>
</alm:attributeList>
```

5.6.4 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].

5.6.5 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].

5.6.6 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].

5.7 Resource: Individual attribute for a contact

The resource used is:

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

The userId, contactId and ResourceRelPath must be percent-encoded according to [RFC3986].

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

5.7.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the user who created a member lists. Example: tel:+1-555-555-0100
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 5.2.6. {name} indicates the name of the attribute. Example: display-name

5.7.2 Response Codes

5.7.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.7.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.7.3 GET

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

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

5.7.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

5.7.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:attribute xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1" name="cellphone">
  <value>tel:+1-555-555-0200</value>
</alm:attribute>
```

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

5.7.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/contacts/alice/attribute/alien HTTP/1.1
Accept: application/xml
Host: example.com
```

5.7.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:55:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <serviceException>
    <messageId>SVC0002</messageId>
```

```

<text>Invalid Invalid input value for message part %1</text>
<variables>Alien</variables>
</serviceException>
</common:requestError>

```

5.7.4 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 member lists that are associated with the contact.

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

5.7.4.1.1 Request

```

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/married HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Host: example.com

```

```

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

```

5.7.4.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Location:http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/married
Date: Fri, 10 Dec 2010 12:51:59 GMT

```

```

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

```

5.7.5 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].

5.7.6 DELETE

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

5.7.6.1 Example: Delete a contact's attribute

(Informative)

5.7.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/display-name HTTP/1.1
Accept: application/xml
Host: example.com
```

5.7.6.1.2 Response

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

5.8 Resource: Member lists

The resource used is:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists?listFilter={listFilterValue}&indivFilter={indivFilterValue}`

The userId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of listFilter and indivFilter query parameters.

5.8.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: <code>http://example.com/exampleAPI</code>
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: <code>tel:+1-555-555-0100</code>
listFilter	defines the level of information related to a list, returned in a GET response body (see 5.4 for details)
indivFilter	defines the level of information related to members of a list, returned in a GET response body (see 5.4 for details)

5.8.2 Response Codes

5.8.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.8.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.8.3 GET

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

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

5.8.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists HTTP/1.1
Accept: application/xml
Host: example.com
```

5.8.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1">
  <memberList memberListId="1234">
    <memberCollection>
      <member memberId="sip:alice@example.com">
        <attributeList>
          <attribute name="display-name">
            <value>Alice</value>
          </attribute>
        </attributeList>
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
      </member>
    </memberCollection>
    <category>URList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com:list=1234</sharedId>
    </sharedListIdentity>
    <attributeList>
      <attribute name="display-name">
        <value>Friends</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
  </memberList>
  <memberList memberListId="5678">
    <memberCollection>
      <member memberId="mailto:liza@example.com">
        <attributeList>
          <attribute name="display-name">
            <value>Wife</value>
          </attribute>
        </attributeList>
      </member>
    </memberCollection>
  </memberList>
</alm:memberListCollection>
```

```

    </attribute>
    <attribute name="age">
      <value>42</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/mailto%3AAliza@example.com</resourceURL>
</member>
<member memberId="mailto:serezha@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Son</value>
    </attribute>
    <attribute name="phone">
      <value>tel:+1-555-555-0260</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/members/mailto%3Aserezha@example.com</resourceURL>
</member>
</memberCollection>
<attributeList>
  <attribute name="display-name">
    <value>Family</value>
  </attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

5.8.3.2 Example 2: Get all member lists belonging to a user without member list attributes and without members (Informative)

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

5.8.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists? listFilter=~noAttr&indivFilter=~none HTTP/1.1
Accept: application/xml
Host: example.com

```

5.8.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">
    <category>URList</category>
    <sharedListIdentity>

```

```

    <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/mailto%3Abob@example.com/memberLists/1234</resourceURL>
</memberList>
<memberList memberListId="5678">
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/mailto%3Abob@example.com/memberLists/5678</resourceURL>
</memberList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/mailto%3Abob@example.com/memberLists </resourceURL>
</alm:memberListCollection>

```

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

5.8.3.3.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists?listFilter=display-name&indivFilter=display-name&indivFilter=phone HTTP/1.1
Accept: application/xml
Host: example.com

```

5.8.3.3.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">
    <memberCollection>
      <member memberId="sip:alice@example.com">
        <attributeList>
          <attribute name="display-name">
            <value>Alice</value>
          </attribute>
        </attributeList>
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
      </member>
    </memberCollection>
    <category>URList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com;list=1234</sharedId>
    </sharedListIdentity>
    <attributeList>
      <attribute name="display-name">
        <value>Friends</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
  </memberList>
</alm:memberListCollection>

```



```

</memberList>
<memberList memberListId="5678">
  <memberCollection>
    <member memberId="mailto:liza@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Wife</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3AAliza@example.com</resourceURL>
    </member>
    <member memberId="mailto:serezha@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Son</value>
        </attribute>
        <attribute name="phone">
          <value>tel:+1-555-555-0260</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/members/mailto%3Aserezha@example.com</resourceURL>
    </member>
  </memberCollection>
  <attributeList>
    <attribute name="display-name">
      <value>Family</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

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

5.8.3.4.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists?listFilter=display-name&indivFilter=~none HTTP/1.1
Accept: application/xml
Host: example.com

```

5.8.3.4.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">

```

```

<category>URLList</category>
<sharedListIdentity>
  <sharedId>sip:bob@example.com;list=1234</sharedId>
</sharedListIdentity>
<attributeList>
  <attribute name="display-name">
    <value>Friends</value>
  </attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</memberList>
<memberList memberListId="5678">
  <attributeList>
    <attribute name="display-name">
      <value>Family</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

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

5.8.3.5.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists?listFilter=display-name&indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.8.3.5.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">
    <memberCollection>
      <member memberId="sip:alice@example.com">
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
      </member>
    </memberCollection>
  </memberList>
</alm:memberListCollection>
<category>URLList</category>
<sharedListIdentity>
  <sharedId>sip:bob@example.com;list=1234</sharedId>
</sharedListIdentity>
<attributeList>
  <attribute name="display-name">
    <value>Friends</value>
  </attribute>
</attributeList>

```

```

</attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</memberList>
<memberList memberListId="5678">
  <memberCollection>
    <member memberId="mailto:liza@example.com">
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aliza@example.com</resourceURL>
    </member>
    <member memberId="mailto:serezha@example.com">
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/members/mailto%3Aserezha@example.com</resourceURL>
    </member>
  </memberCollection>
  <attributeList>
    <attribute name="display-name">
      <value>Family</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

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

5.8.3.6.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists?listFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.8.3.6.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">
    <memberCollection>
      <member memberId="sip:alice@example.com">
        <attributeList>
          <attribute name="display-name">
            <value>Alice</value>
          </attribute>
        </attributeList>
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-

```

```

0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
  </member>
</memberCollection>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</memberList>
<memberList memberListId="5678">
  <memberCollection>
    <member memberId="mailto:liza@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Wife</value>
        </attribute>
        <attribute name="age">
          <value>42</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/mailto%3Aliza@example.com</resourceURL>
    </member>
    <member memberId="mailto:serezha@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Son</value>
        </attribute>
        <attribute name="phone">
          <value>tel:+1-555-555-0260</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/members/mailto%3Aserezha@example.com</resourceURL>
    </member>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

5.8.4 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].

5.8.5 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].

5.8.6 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].

5.9 Resource: Individual member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}?listFilter={listFilterValue}&indivFilter={indivFilterValue}

The userId and memberListId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of listFilter and indivFilter query parameters.

5.9.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: 1234
listFilter	defines the level of information related to a list, returned in a GET response (see 5.4 for details)
indivFilter	defines the level of information related to members of a list, returned in a GET response (see 5.4 for details)

5.9.2 Response Codes

5.9.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.9.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.9.3 GET

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

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

5.9.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234 HTTP/1.1
```

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

5.9.3.1.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList memberListId="1234" xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Alice</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <attributeList>
    <attribute name="display-name">
      <value>Friends</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</alm:memberList>

```

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

5.9.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234?listFilter=~noAttr&indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.9.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList memberListId="1234" xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/mailto%3Aalice@example.com/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>

```

```
</alm:memberList>
```

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

5.9.3.3.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234?indivFilter=display-name HTTP/1.1
Accept: application/xml
Host: example.com
```

5.9.3.3.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList memberListId="1234" xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Alice</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</alm:memberList>
```

5.9.3.4 Example 4: Retrieve a non existing member list (Informative)

5.9.3.4.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/4321 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.9.3.4.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

```

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="MemberList"
    href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/4321"/>
  <serviceException>
    <messageId>SVC0002</messageId>
    <text>Invalid input value for message part %1</text>
    <variables>memberListId</variables>
  </serviceException>
</common:requestError>

```

5.9.4 PUT

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

5.9.4.1 Example: Create a member list

(Informative)

5.9.4.1.1 Request

```

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/bobPublic HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Host: example.com

```

```

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberListId="bobPublic">
  <memberCollection>
    <member memberId="mailto:alice@example.com"/>
  </memberCollection>
</alm:memberList>

```

5.9.4.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/bobPublic
Date: Fri, 10 Dec 2010 12:51:59 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberListId="bobPublic">
  <memberCollection>
    <member memberId="mailto:alice@example.com"/>
  </memberCollection>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/bobPublic</resourceURL>
</alm:memberList>

```


5.9.5 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].

5.9.6 DELETE

This operation deletes a contact list.

5.9.6.1 Example: Delete a member list (Informative)

5.9.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/bobPublic HTTP/1.1
Accept: application/xml
Host: example.com
```

5.9.6.1.2 Response

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

5.10 Resource: Attributes for a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/attributes

The userId and memberListId must be percent-encoded according to [RFC3986].

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

5.10.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100

5.10.2 Response Codes

5.10.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.10.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.10.3 GET

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

5.10.3.1 Example: Get all attributes for a member lists belonging to Bob (Informative)

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

5.10.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes HTTP/1.1
Accept: application/xml
Host: example.com
```

5.10.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:attributeList xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1">
  <attribute name="display-name">
    <value>Friends</value>
  </attribute>
  <attribute name="date-created">
    <value>Fri, 10 Dec 2010 11:28:34 GMT</value>
  </attribute>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes</resourceURL>
</alm:attributeList>
```

5.10.4 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].

5.10.5 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].

5.10.6 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].

5.11 Resource: Individual attribute for a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/attributes/[ResourceRelPath]

The userId, membermemberListId and ResourceRelPath must be percent-encoded according to [RFC3986].

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

5.11.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member 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 5.2.6. {name} indicates the name of the attribute. Example: display-name

5.11.2 Response Codes

5.11.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.11.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.11.3 GET

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

5.11.3.1 Example 1: Retrieve an attribute value

(Informative)

5.11.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/date-created HTTP/1.1
Accept: application/xml
Host: example.com
```

5.11.3.1.2 Response

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

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

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

5.11.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/alien HTTP/1.1
Accept: application/xml
Host: example.com
```

5.11.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT

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

5.11.4 PUT

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

5.11.4.1 Example: Create an attribute (Informative)

5.11.4.1.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/display-name HTTP/1.1
Content-Type: application/xml
Content-Length: nnnn
Accept: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<alm:attribute xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" name="display-name">
```

```
<value>Close friends</value>
</alm:attribute>
```

5.11.4.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location:http://example.com/exampleAPI/1/ addresslistmgt/tel%3A%2B1-555-555-0123/memberLists/1234/attributes/display-name
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

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

5.11.5 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].

5.11.6 DELETE

This operation deletes a member list.

5.11.6.1 Example: Delete an attribute (Informative)

5.11.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/display-name HTTP/1.1Accept:
application/xml
Host: example.com
```

5.11.6.1.2 Response

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

5.12 Resource: Members in a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/?indivFilter={indivFilterValue}

The userId and memberListId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of indivFilter query parameter.

5.12.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: myList
indivFilter	defines the level of information related to members of a member list, returned in a GET response body (see 5.4 for details)

5.12.2 Response Codes

5.12.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.12.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.12.3 GET

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

5.12.3.1 Example 1: Get all members for a member list belonging to a user with all attributes (default) (Informative)

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

5.12.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members HTTP/1.1
Accept: application/xml
Host: example.com
```

5.12.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <member memberId="mailto:alice@example.com">
```

```

<attributeList>
  <attribute name="display-name">
    <value>Alice</value>
  </attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/mailto%3Aalice@example.com
</resourceURL>
</member>
<member memberId="tel:+1-555-555-0124">
  <attributeList>
    <attribute name="display-name">
      <value>Bob</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/tel%3A%2B1-555-555-0124
  </resourceURL>
</member>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members</resourceURL>
</alm:memberCollection>

```

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

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

5.12.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members?indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.12.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <member memberId="mailto:alice@example.com">
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/mailto%3Aalice@example.com
    </resourceURL>
  </member>
  <member memberId="tel:+1-555-555-0124">
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/1234/members/tel%3A%2B1-555-555-0124</resourceURL>
  </member>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-

```

```
0100/memberLists/1234/members</resourceURL>
</alm:memberCollection>
```

5.12.4 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].

5.12.5 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].

5.12.6 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].

5.13 Resource: Individual member in a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}?indivFilter={indivFilterValue}

The userId, memberListId and memberId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of indivFilter query parameter.

5.13.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: 6789
memberId	member identifier. Example: mailto:alice@example.com
indivFilter	defines the level of information related to members of a member list, returned in a GET response body (see 5.4 for details). indivFilter=~none and indivFilter=~noAttr are not supported for this resource.

5.13.2 Response Codes

5.13.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.13.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.13.3 GET

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

5.13.3.1 Example 1: Retrieve a member (Informative)

5.13.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com HTTP/1.1
Accept: application/xml
Host: example.com
```

5.13.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="sip:alice@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Alice</value>
    </attribute>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0155</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com
</resourceURL>
</alm:member>
```

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

5.13.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

5.13.3.2.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="sip:alice@example.com">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0155</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com</resourceURL>
</alm:member>
```

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

5.13.3.3.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aajoe@example.com
HTTP/1.1 Accept: application/xml
Host: example.com
```

5.13.3.3.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">

  <serviceException>
    <messageId>SVC0002</messageId>
    <text> Invalid input value for message part %1</text>
    <variables>memberId</variables>
  </serviceException>
</common:requestError>
```

5.13.4 PUT

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

5.13.4.1 Example 1: Create a member without a link to the Contact (Informative)

5.13.4.1.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3Aole@example.com HTTP/1.1
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="mailto:ole@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Ole</value>
    </attribute>
  </attributeList>
</alm:member>
```

5.13.4.1.2 Response

HTTP/1.1 201 Created
 Content-Type: application/xml
 Content-Length: nnnn
 Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3A:ole@example.com
 Date: Fri, 10 Dec 2010 12:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="mailto:ole@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Ole</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3Aole@example.com
  </resourceURL>
</alm:member>
```

5.13.4.2 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 exist.

5.13.4.2.1 Request

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3A:joe@example.com HTTP/1.1
 Accept: application/xml
 Content-Type: application/xml
 Host: example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="mailto:joe@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Joe</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
```

```
0100/memberLists/6789/members/mailto%3Ajoe@example.com
</resourceURL>
<link rel="Contact" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/joe"/>
</alm:member>
```

5.13.4.2.2 Response

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

5.13.5 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].

5.13.6 DELETE

This operation deletes a member from a member list.

5.13.6.1 Example: Delete a member (Informative)

5.13.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/tel%3A%2B1-555-555-0125
HTTP/1.1
Accept: application/xml
Host: example.com
```

5.13.6.1.2 Response

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

5.14 Resource: Attributes for a member in a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}/attributes

The userId, memberListId and memberId must be percent-encoded according to [RFC3986].

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

5.14.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: myList.
memberId	member identifier. Example: tel:+1-555-555-0126

5.14.2 Response Codes

5.14.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.14.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.14.3 GET

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

5.14.3.1 Example: Get all attributes of a member of a member list (Informative)

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

5.14.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3Aalice@example.com/attributes
HTTP/1.1
Accept: application/xml
Host: example.com
```

5.14.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:attributeList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <attribute name="display-name">
    <value>Alice</value>
  </attribute>
  <attribute name="cellphone">
    <value>tel:+1-555-555-0155</value>
  </attribute>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com/attributes</resourceURL>
</alm:attributeList>
```

5.14.4 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].

5.14.5 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].

5.14.6 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].

5.15 Resource: Individual attribute for a member in a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/members/{memberId}/attributes/[ResourceRelPath]

The userId, memberListId, memberId, and ResourceRelPath must be percent-encoded according to [RFC3986].

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

5.15.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: myFriends
memberId	member identifier. Example: tel:+1-555-555-0126
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this light-weight resource is {name}, as defined in the [ResourceRelPath] column in table 5.2.6. {name} indicates the name of the attribute. Example: display-name

5.15.2 Response Codes

5.15.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.15.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.15.3 GET

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

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

5.15.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/sip%3Aalice@example.com/attributes/display-name HTTP/1.1
Accept: application/xml
Host: example.com
```

5.15.3.1.2 Response

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

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

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

5.15.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/sip%3Aalice@example.com/attributes/pet HTTP/1.1
Accept: application/xml
Host: example.com
```

5.15.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:55:59 GMT

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

5.15.4 PUT

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

5.15.4.1 Example: Create a member's attribute

(Informative)

5.15.4.1.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/6789/members/sip%3Aalice@example.com/attributes/born HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Host: example.com

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

5.15.4.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Location:http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/6789/members/sip:alice@example.com/attributes/born
Date: Fri, 10 Dec 2010 12:51:59 GMT

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

5.15.5 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].

5.15.6 DELETE

This operation deletes a member in a member list.

5.15.6.1 Example: Delete an attribute

(Informative)

5.15.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/6789/members/sip%3Aalice@example.com/attributes/born HTTP/1.1
Accept: application/xml
Host: example.com
```


5.15.6.1.2 Response

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

5.16 Resource: Member list references

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/memberListReferences

The userId and memberListId must be percent-encoded according to [RFC3986].

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

5.16.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
memberListId	member list identifier. Example: myList

5.16.2 Response Codes

5.16.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.16.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.16.3 GET

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

5.16.3.1 Example: Get all member list references in a member lists (Informative)

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

5.16.3.1.1 Request

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/allFriends/memberListReferences HTTP/1.1
Accept: application/xml
Host: example.com

5.16.3.1.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListReferenceCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/memberLists/friends/memberListReferences</resourceURL>
  <link rel="MemberList" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234"/>
  <link rel="MemberList" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678"/>
</alm:memberListReferenceCollection>

```

5.16.4 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].

5.16.5 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].

5.16.6 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].

5.17 Resource: Individual member list reference in a member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/{memberListId}/memberListReferences/[ResourceRelPath]

The userId, memberListId, and ResourceRelPath must be percent-encoded according to [RFC3986].

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

5.17.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	identity of the user who created a member lists. Example: tel:+1-555-555-0100
memberListId	member 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 5.2.10. {href} indicates the link to the referenced member list.
-------------------	--

5.17.2 Response Codes

5.17.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.17.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.17.3 GET

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

5.17.3.1 Example 1: Retrieve a member list reference (Informative)

5.17.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberlist/allFriends/memberListReferences/http%3A/example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.17.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:link xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" rel="MemberList"
href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234"/>
```

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

5.17.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberlist/allFriends/memberListReferences/http%3A/example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1235 HTTP/1.1
HTTP/1.1
Accept: application/xml
Host: example.com
```

5.17.3.2.2 Response

HTTP/1.1 404 Not Found
 Content-Type: application/xml
 Content-Length: nnn
 Date: Fri, 10 Dec 2010 12:55:59 GMT

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

5.17.4 PUT

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

5.17.4.1 Example: Create a member list reference (Informative)

5.17.4.1.1 Request

PUT /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberlist/allFriends/memberListReferences/http%3A/example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/5678 HTTP/1.1
 Content-Type: application/xml
 Accept: application/xml
 Host: example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:link xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1" rel="MemberList"
href="http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/5678"/>
```

5.17.4.1.2 Response

HTTP/1.1 201 Created
 Content-Type: application/xml
 Location: http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberlist/allFriends/memberListReferences/http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/5678
 Date: Fri, 10 Dec 2010 12:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:link xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1" rel="MemberList"
href="http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/5678"/>
```

5.17.5 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].

5.17.6 DELETE

This operation deletes a member list reference in a member list.

5.17.6.1 Example: Delete a member list reference (Informative)

5.17.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/allFriends/memberListReferences/
      http%3A//example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.17.6.1.2 Response

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

5.18 Resource: List changes subscriptions

The resource used is:

`http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/subscriptions/listChanges`

The `userId` must be percent-encoded according to [RFC3986].

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

5.18.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: <code>http://example.com/exampleAPI</code>
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: <code>tel:+1-555-555-0100</code>

5.18.2 Response Codes

5.18.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.18.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.18.3 GET

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

5.18.3.1 Example: Get member list changes subscriptions (Informative)

5.18.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges HTTP/1.1
Accept: application/xml
Host: example.com
```

5.18.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscriptionCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <listChangesSubscription>
    <memberListId>1234</memberListId>
    <callbackReference>
      <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <clientCorrelator>123</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>1958</duration>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321</resourceURL>
  </listChangesSubscription>
  <listChangesSubscription>
    <anyContacts/>
    <callbackReference>
      <notifyURL>http://client.example.com/notifications/contacts</notifyURL>
      <callbackData>54321</callbackData>
    </callbackReference>
    <clientCorrelator>456</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <duration>2641</duration>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/123456789</resourceURL>
  </listChangesSubscription>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges</resourceURL>
</alm:listChangesSubscriptionCollection>
```

5.18.4 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].

5.18.5 POST

This operation is used for creation of the new subscription.

5.18.5.1 Example: Create new subscription for member list changes notification (Informative)

5.18.5.1.1 Request

```
POST /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com
```

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberListId>1234</memberListId>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
</alm:listChangesSubscription>
```

5.18.5.1.2 Response

```
HTTP/1.1 201 Created
```

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

```
Content-Length: nnnn
```

```
Date: Fri, 10 Dec 2010 21:33:52 GMT
```

```
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321
```

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberListId>1234</memberListId>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321</resourceURL>
</alm:listChangesSubscription>
```

5.18.6 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].

5.19 Resource: Individual member list changes subscription

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/subscriptions/listChanges/{subscriptionId}

The userId must be percent-encoded according to [RFC3986].

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

5.19.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
subscriptionId	identifier associated with this specific subscription

5.19.2 Response Codes

5.19.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.19.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.19.3 GET

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

5.19.3.1 Example: Get member list changes subscription (Informative)

Retrieve a specific member list changes subscription.

5.19.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.19.3.1.2 Response

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



```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberListId>1234</memberListId>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>6817</duration>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321</resourceURL>
</alm:listChangesSubscription>
```

5.19.4 PUT

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

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

5.19.4.1.1 Request

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321 HTTP/1.1

Content-Type: application/xml

Accept: application/xml

Host: example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberListId>1234</memberListId>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321</resourceURL>
</alm:listChangesSubscription>
```

5.19.4.1.2 Response

HTTP/1.1 200 OK

Content-Type: application/xml

Date: Fri, 10 Dec 2010 12:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberListId>1234</memberListId>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/memberLists/1234</notifyURL>
    <callbackData>12345</callbackData>
```

```
</callbackReference>
<clientCorrelator>123</clientCorrelator>
<applicationTag>myApp</applicationTag>
<duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/subscriptions/listChanges/987654321</resourceURL>
</alm:listChangesSubscription>
```

5.19.5 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].

5.19.6 DELETE

This operation deletes a subscription to member list changes.

5.19.6.1 Example: Delete a member list changes subscription (Informative)

5.19.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.19.6.1.2 Response

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

5.20 Resource: Client resource for member list changes notifications

This resource is a client provided callback URL for client notification about outbound message delivery status. ParlayREST does not make any assumption about the structure of this URL.

5.20.1 Request URI variables

Client provided.

5.20.2 Response Codes

5.20.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.20.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.20.3 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].

5.20.4 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].

5.20.5 POST

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

5.20.5.1 Example 1: Notification for subscription with sendFullMemberListContent=true (default) (Informative)

5.20.5.1.1 Request

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

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangeNotification xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <callbackData>12345</callbackData>
  <resourceStatus>Active</resourceStatus>
  <duration>3480</duration>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/
  members/sip:alice@example.com"/>
  <link rel="ListChangesSubscription" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/
  listChanges/987654321"/>
</alm:listChangeNotification>
```

5.20.5.1.2 Response

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

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

5.20.5.2.1 Request

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

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangeNotification xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
```

```

<callbackData>54321</callbackData>
<resourceStatus>Active</resourceStatus>
<duration>4217</duration>
<link rel="Contact" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/ole"/>
<link rel="ListChangesSubscription" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/123456789"/>
</alm:listChangeNotification>

```

5.20.5.2.2 Response

```

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

```

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

5.20.5.3.1 Request

```

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

```

```

<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangeNotification xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <callbackData>54321</callbackData>
  <resourceStatus>TerminatedTimeout</resourceStatus>
  <link rel="ListChangesSubscription" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/123456789"/>
</alm:listChangeNotification>

```

5.20.5.3.2 Response

```

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

```

5.20.6 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].

5.21 Resource: Collection of contacts shared by other user

The resource used is:

```

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/contacts?listFilter={listFilterValue}&indivFilter={indivFilterValue}

```

The `userId` and `otherUserId` must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of `listFilter` and `indivFilter` query parameters.

5.21.1 Request URI variables

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

Name	Description
<code>serverRoot</code>	server base url: hostname+port+base path. Example: <code>http://example.com/exampleAPI</code>
<code>apiVersion</code>	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
<code>userId</code>	requesting user identifier. Example: <code>tel:+1-555-555-0100</code>
<code>otherUserId</code>	sharing user identifier. Example: <code>tel:+1-555-555-0103</code>
<code>listFilter</code>	defines the level of information related to a list, returned in a GET response body (see 5.4 for details)
<code>indivFilter</code>	defines the level of information related to members of a list, returned in a GET response body (see 5.4 for details)

5.21.2 Response Codes

5.21.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.21.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.21.3 GET

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

5.21.3.1 Example 1: Get contacts shared by a user identified by otherUserId (Informative)

5.21.3.1.1 Request – all contact attributes (default)

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts HTTP/1.1
Accept: application/xml
Host: example.com
```

5.21.3.1.2 Response

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

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

```

<alm:contactCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <contact contactId="alice">
    <sharedIdentity>
      <sharedId>sip:alice@example.com</sharedId>
    </sharedIdentity>
    <attributeList>
      <attribute name="display-name">
        <value>Alice</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/sip:alice@example.com</resourceURL>
  </contact>
  <contact contactId="ole">
    <sharedIdentity>
      <sharedId>mailto:ole@example.com</sharedId>
    </sharedIdentity>
    <attributeList>
      <attribute name="display-name">
        <value>Ole</value>
      </attribute>
    </attributeList>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/mailto%3Aole@example.com</resourceURL>
    <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/shared/memberLists/6789/members/mailto%3Aole@example.com"/>
  </contact>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts</resourceURL>
</alm:contactCollection>

```

5.21.3.2 Example 2: Get contact lists shared by a user identified by otherUserId without contact lists attributes and without member attributes (Informative)

5.21.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts}?listFilter=~noAttr&indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.21.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contactCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <contact contactId="alice">
    <sharedIdentity>
      <sharedId>sip:alice@example.com</sharedId>
    </sharedIdentity>

```

```

<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/sip:alice@example.com</resourceURL>
</contact>
<contact contactId="ole">
  <sharedIdentity>
    <sharedId>mailto:ole@example.com</sharedId>
  </sharedIdentity>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/mailto%3Aole@example.com</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/shared/memberLists/6789/members/mailto%3Aole@example.com"/>
</contact>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts</resourceURL>
</alm:contactCollection>

```

5.21.4 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].

5.21.5 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].

5.21.6 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].

5.22 Resource: Individual shared contact

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/shared/{otherUserId}/contacts/{sharedIdentity}?listFilter={listFilterValue}&indivFilter={indivFilterValue}

The userId, otherUserId and sharedIdentity must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of indivFilter query parameters.

5.22.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI

apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0260
otherUserId	user identifier. Example: tel:+1-555-555-0100
sharedIdentity	identifier of the shared contact. Example: alice or mailto:alice@example.com
listFilter	defines the level of information related to a list, returned in a GET response body (see 5.4 for details)
indivFilter	defines the level of information related to members of a list, returned in a GET response body (see 5.4 for details)

5.22.2 Response Codes

5.22.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.22.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.22.3 GET

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

5.22.3.1 Example 1: Retrieve information about individual shared contact (Informative)

In this example the contactId is known to the requestor.

5.22.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B 1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/alice HTTP/1.1
Accept: application/xml
Host: example.com
```

5.22.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="alice">
  <attributeList>
<attribute name="display-name">
  <value>Alice</value>
</attribute>
<attribute name="cellphone">
  <value>tel:+1-555-555-0200</value>
</attribute>
```



```

<attribute name="state">
  <value>California</value>
</attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/alice</resourceURL>
<link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
</alm:contact>

```

5.22.3.2 Example 2: Retrieve information about individual shared contact (Informative)

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

5.22.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/alice HTTP/1.1
Accept: application/xml
Host: example.com

```

5.22.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" contactId="alice">
  <attributeList>
<attribute name="display-name">
  <value>Alice</value>
  </attribute>
<attribute name="cellphone">
  <value>tel:+1-555-555-0200</value>
  </attribute>
<attribute name="state">
  <value>California</value>
  </attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/alice</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
</alm:contact>

```

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

5.22.3.3.1 Request

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/alice?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

5.22.3.3.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:contact xmlns:alm="urn:oma:xml:rest:addresslistmgmt:1" contactId="alice">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0200</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/alice</resourceURL>
  <link rel="Member" href="http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122"/>
</alm:contact>
```

5.22.4 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].

5.22.5 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].

5.22.6 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].

5.23 Resource: Collection of shared member lists

The resource used is:

```
http://{serverRoot}/{apiVersion}/addresslistmgmt/{userId}/shared/{otherUserId}/memberLists?listFilter={~noAttr | name1, name2, ...}&indivFilter={~none | ~noAttr | name1, name2, ...}
```

The userId and otherUserId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of listFilter and indivFilter query parameters.

5.23.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	requesting user identifier. Example: tel:+1-555-555-0100
otherUserId	sharing user identifier. Example: tel:+1-555-555-0103
listFilter	defines the level of information related to a list, returned in a GET response body (see 5.8 for details)
indivFilter	defines the level of information related to members of a list, returned in a GET response body (see 5.8 for details)

5.23.2 Response Codes

5.23.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.23.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.23.3 GET

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

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

5.23.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists HTTP/1.1
Accept: application/xml
Host: example.com
```

5.23.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
```

```

<memberList memberListId="1234">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Alice</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <category>URLList</category>
  <sharedListIdentity>
    <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
    <attribute name="display-name">
      <value>Friends</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</memberList>
<memberList memberListId="5678">
  <memberCollection>
    <member memberId="mailto:liza@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Wife</value>
        </attribute>
        <attribute name="age">
          <value>42</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678/members/mailto%3Aliza@example.com</resourceURL>
    </member>
    <member memberId="mailto:serezha@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Son</value>
        </attribute>
        <attribute name="phone">
          <value>tel:+1-555-555-0260</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678/members/mailto%3Aserezha@example.com</resourceURL>
    </member>
  </memberCollection>
  <attributeList>
    <attribute name="display-name">
      <value>Family</value>
    </attribute>
  </attributeList>

```

```

<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

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

5.23.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists?listFilter=~noAttr&indivFilter=phone HTTP/1.1
Accept: application/xml
Host: example.com

```

5.23.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListCollection xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <memberList memberListId="1234">
    <memberCollection>
      <member memberId="sip:alice@example.com">
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
      </member>
    </memberCollection>
    <category>URIList</category>
    <sharedListIdentity>
      <sharedId>sip:bob@example.com;list=1234</sharedId>
    </sharedListIdentity>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
  </memberList>
  <memberList memberListId="5678">
    <memberCollection>
      <member memberId="mailto:liza@example.com">
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678/members/mailto%3Aliza@example.com</resourceURL>
      </member>
      <member memberId="mailto:serezha@example.com">
        <attributeList>
          <attribute name="phone">
            <value>tel:+1-555-555-0260</value>
          </attribute>
        </attributeList>
        <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-

```

```

0100/memberLists/5678/members/mailto%3Asereza@example.com</resourceURL>
  </member>
</memberCollection>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-
0100/memberLists/5678</resourceURL>
</memberList>
</alm:memberListCollection>

```

5.23.4 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].

5.23.5 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].

5.23.6 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].

5.24 Resource: Individual shared member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgmt/{userId}/shared/{otherUserId}/memberLists/{sharedListIdentity}?listFilter={~noAttr | name1, name2, ...}&indivFilter={~none | ~noAttr | name1, name2, ...}

The userId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of listFilter and indivFilter query parameters.

5.24.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	requesting user identifier. Example: tel:+1-555-555-0100
otherUserId	sharing user identifier. Example: tel:+1-555-555-0103
sharedMemberListId	shared member list identifier. Example: myList
listFilter	defines the level of information related to a list, returned in a GET response body (see 5.8 for details)

indivFilter	defines the level of information related to members of a list, returned in a GET response body (see 5.8 for details)
-------------	--

5.24.2 Response Codes

5.24.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.24.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.24.3 GET

This operation is used for retrieval of all single member list information for a given user identity.

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

5.24.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.24.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberListId="1234">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <attributeList>
        <attribute name="display-name">
          <value>Alice</value>
        </attribute>
      </attributeList>
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <category>URLList</category>
  <sharedListIdentity>
    <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
    <attribute name="display-name">
```

```

    <value>Friends</value>
  </attribute>
</attributeList>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</alm:memberList>

```

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

5.24.3.2.1 Request

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234?indivFilter=~noAttr HTTP/1.1
Accept: application/xml
Host: example.com

```

5.24.3.2.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberListId="1234">
  <memberCollection>
    <member memberId="sip:alice@example.com">
      <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
    </member>
  </memberCollection>
  <category>URList</category>
  <sharedListIdentity>
    <sharedId>sip:bob@example.com;list=1234</sharedId>
  </sharedListIdentity>
  <attributeList>
    <attribute name="display-name">
      <value>Friends</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234</resourceURL>
</alm:memberList>

```

5.24.4 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].

5.24.5 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].

5.24.6 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].

5.25 Resource: Individual member information from shared member list

The resource used is:

http://{serverRoot}/{apiVersion}/addresslistmgt/{userId}/memberLists/sharedBy/{otherUserId}/{sharedMemberListId}/members/{sharedMemberId}...}&indivFilter={name1, name2, ...}

The userId, otherUserId and sharedMemberId must be percent-encoded according to [RFC3986].

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

Note: see 5.4 for the definition of indivFilter query parameter.

5.25.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	requesting user identifier. Example: tel:+1-555-555-0100
otherUserId	other user identifier. Example: tel:+1-555-555-0103
sharedListIdentity	shared member list identifier. Example: sip:alice@example.com;list=friends
memberId	member list member identifier. Example: mailto:alice@example.com
indivFilter	defines the level of information related to members of a list, returned in a GET response body (see 5.4 for details). indivFilter=~none and indivFilter=~noAttr are not supported for this resource.

5.25.2 Response Codes

5.25.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.25.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.25.3 GET

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

5.25.3.1 Example 1: Get individual member information about user identified as memberId from the list shared by a user identified by otherUserId (default) (Informative)

5.25.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com HTTP/1.1
Accept: application/xml
Host: example.com
```

5.25.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="sip:alice@example.com">
  <attributeList>
    <attribute name="display-name">
      <value>Alice</value>
    </attribute>
    <attribute name="cellphone">
      <value>1-555-555-0105</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com</resourceURL>
</alm:member>
```

5.25.3.2 Example 2: Get individual member information about user identified as memberId from the list shared by a user identified by otherUserId (default) (Informative)

5.25.3.2.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip%3Aalice@example.com?indivFilter=cellphone HTTP/1.1
Accept: application/xml
Host: example.com
```

5.25.3.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```

Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<alm:member xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" memberId="sip:alice@example.com">
  <attributeList>
    <attribute name="cellphone">
      <value>tel:+1-555-555-0105</value>
    </attribute>
  </attributeList>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip%3Aalice@example.com</resourceURL>
</alm:member>

```

5.25.4 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].

5.25.5 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].

5.25.6 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].

5.26 Resource: Authorization rules

The resource used is:

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

The userId must be percent-encoded according to [RFC3986].

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

5.26.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100

5.26.2 Response Codes

5.26.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.26.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.26.3 GET

This operation is used for retrieval of all authorization rules.

5.26.3.1 Example: Get authorization rules

(Informative)

5.26.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules HTTP/1.1
Accept: application/xml
Host: example.com
```

5.26.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:ruleList xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <rule ruleName="allowAllContacts">
    <clientCorrelator>123</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <applToContacts>
      <allContacts/>
    </applToContacts>
    <authorizedTo>
      <memberListId>1234</memberListId>
    </authorizedTo>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1</resourceURL>
  </rule>
  <rule ruleName="allowOwnContact">
    <clientCorrelator>456</clientCorrelator>
    <applicationTag>myApp</applicationTag>
    <applToContacts>
      <contactId>bob</contactId>
    </applToContacts>
    <authorizedTo>
      <allSharedIdentities/>
    </authorizedTo>
    <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule2</resourceURL>
  </rule>
  <rule ruleName="allowFriendsList">
    <clientCorrelator>789</clientCorrelator>
```

```

<applicationTag>myApp</applicationTag>
<applToMemLists>
  <memberListId>1234</memberListId>
</applToMemLists>
<authorizedTo>
  <memberListId>1234</memberListId>
</authorizedTo>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3</resourceURL>
</rule>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules</resourceURL>
</alm:ruleList>

```

5.26.4 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].

5.26.5 POST

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

5.26.5.1 Example: Create new rule for memberList and referenced contacts (Informative)

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

5.26.5.1.1 Request

```

POST /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules HTTP/1.1
Accept: application/xml
Host: example.com

```

```

<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="combinedRule">
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToMemListsAndContacts>
    <memberListId>5678</memberListId>
  </applToMemListsAndContacts>
  <authorizedTo>
    <memberListId>5678</memberListId>
  </authorizedTo>
</alm:rule>

```

5.26.5.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnn
Date: Fri, 10 Dec 2010 21:33:52 GMT
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3

```

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="combinedRule">
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToMemListsAndContacts>
    <memberListId>5678</memberListId>
  </applToMemListsAndContacts>
  <authorizedTo>
    <memberListId>5678</memberListId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3</resourceURL>
</alm:rule>
```

5.26.6 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].

5.27 Resource: Individual authorization rule

The resource used is:

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

The userId and ruleId must be percent-encoded according to [RFC3986].

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

5.27.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
ruleId	rule identifier: Example: 987654321

5.27.2 Response Codes

5.27.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.27.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.27.3 GET

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

5.27.3.1 Example: Get authorization rule (Informative)

5.27.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule1 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.27.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="allowAllContacts">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
  <authorizedTo>
    <memberListId>1234</memberListId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1</resourceURL>
</alm:rule>
```

5.27.4 PUT

This operation is used to update an individual authorization rule.

5.27.4.1 Example: Update an existing authorization rule (Informative)

5.27.4.1.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule1 HTTP/1.1
Content-Type: application/xml
Content-Length : nnnn
Accept: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="allowAllContacts">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
```

```

<authorizedTo>
  <memberListId>1234</memberListId>
  <memberListId>5678</memberListId>
</authorizedTo>
<resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1</resourceURL>
</alm:rule>

```

5.27.4.1.2 Response

```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="allowAllContacts">
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToContacts>
    <allContacts/>
  </applToContacts>
  <authorizedTo>
    <memberListId>1234</memberListId>
    <memberListId>5678</memberListId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1</resourceURL>
</alm:rule>

```

5.27.5 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].

5.27.6 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].

5.27.6.1.1 Request

```

DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2 HTTP/1.1
Accept: application/xml
Host: example.com

```

5.27.6.1.2 Response

```

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

```


5.28 Resource: Individual authorization rule data

The resource used is:

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

The userId, ruleId and ResourceRelPath must be percent-encoded according to [RFC3986].

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.

5.28.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)
userId	user identifier. Example: tel:+1-555-555-0100
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 5.2.19.

5.28.2 Response Codes

5.28.2.1 Response Codes

For HTTP response codes, see [OMA_REST_TS_Common].

5.28.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Address List Management, see section 6.

5.28.3 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.

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

5.28.3.1.1 Request

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule1/memberLists/1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

5.28.3.1.2 Response

```
HTTP/1.1 200 OK
```

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

<?xml version="1.0" encoding="UTF-8"?>
<alm:memberListId xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">1234</alm:memberListId>
```

5.28.4 PUT

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

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

5.28.4.1.1 Request

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2/users/mailto%3Aalice@example.com
HTTP/1.1
Content-Type: application/xml
Content-Length : nnnn
Accept: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<alm:userld xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">mailto:alice@example.com</alm:userld>
```

5.28.4.1.2 Response

```
HTTP/1.1 201 Created
Location: /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2/users/mailto%3Aalice@example.com
Content-Type: application/xml
Date: Fri, 10 Dec 2010 12:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<alm:userld xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">mailto:alice@example.com</alm:userld>
```

5.28.5 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].

5.28.6 DELETE

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

5.28.6.1 Example: Delete an authorized identity (Informative)

5.28.6.1.1 Request

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2/users/mailto%3Aalice@example.com
HTTP/1.1
Accept: application/xml
Host: example.com
```

5.28.6.1.2 Response

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

6. Fault definitions

6.1 Service Exceptions

The following Service Exception codes are defined for the ParlayREST Address List Management in addition to those in [3GPP 29.199-13].

6.1.1 SVC0240: Key property changes not allowed

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

6.2 Policy Exceptions

The following Policy Exception codes are defined for the ParlayREST Address List Management in addition to those in [3GPP 29.199-13].

6.2.1 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 Versions: OMA-TS-ParlayREST_AddressListManagement-V1_0	15 Apr 2010	Many	TS skeleton prepared
	19 May 2010	5.1 5.2 5.3	OMA-ARC-REST-2010-0228R03- CR_Address_List_Management_resources_summary OMA-ARC-REST-2010-0229R02- CR_Address_List_Management_data_structures OMA-ARC-REST-2010-0230R02- CR_Address_List_Management_sequence_diagrams
	29 Jun 2010	5.1, 5.2 2.1, 4.1 2.1, 5.4 5.5 5.6	OMA-ARC-REST-2010-0278R04-CR_Add_list_reference_management OMA-ARC-REST-2010-0291R01-CR_ALM_TS_updates_sections_2to5 OMA-ARC-REST-2010-0274R01-CR_ALM_example_1 OMA-ARC-REST-2010-0275R01-CR_ALM_example_2 OMA-ARC-REST-2010-0318R01-CR_ALM_example_3
	7 Jul 2010	5.1 5.3.1 Add 5.7 Add 5.8 Add 5.9 Add 5.10 Add 5.11 Add 5.12 Add B1.x	OMA-ARC-REST-2010-0278R04-CR_Add_list_reference_management.zip OMA-ARC-REST-2010-0305R02- CR_ALM_TS_sequence_diagrams_updates.zip OMA-ARC-REST-2010-0319R02-CR_ALM_example_4 OMA-ARC-REST-2010-0320R04-CR_ALM_example_5 OMA-ARC-REST-2010-0321R03-CR_ALM_example_6 OMA-ARC-REST-2010-0323R02-CR_ALM_example_7 OMA-ARC-REST-2010-0326R03-CR_ALM_example_8 OMA-ARC-REST-2010-0327R02-CR_ALM_example_9.zip OMA-ARC-REST-2010-0335R01-CR_ALM_SCR_update
	14 Jul 2010	5.4+	OMA-ARC-REST-2010-0355-CR_ALM_Validation_errors
	28 Aug 2010	Add app. D Update app. B Replace app. C 5.3.1 D 5.2.x	OMA-ARC-REST-2010-0401-CR_JSON_examples_for_ALM OMA-ARC-REST-2010-0402-CR_SCR_updates_for_ALM OMA-ARC-REST-2010-0432R01- CR_empty_appendix_C_form_urlencoding_in_ALM OMA-ARC-REST-2010-0437- CR_ALM_TS_sequence_diagrams_more_updates OMA-ARC-REST-2010-0394-INP_Fixing_JSON_references REST-2010-A085, OMA-ARC-REST-2010-0445- CR_DevCap_resourceURL_description

Document Identifier	Date	Sections	Description
	27 Sep 2010	N/A 5 E N/A 5 All D	REST-2010-A063, no POST REST-2010-A068 REST-2010-A083, OMA-ARC-REST-2010-0491R01- CR_ALM_implement_AI_from_431R02 REST-2010-A096 REST-2010-A098 REST-2010-A102, OMA-ARC-REST-2010-0516R01- CR_Presence_update_references_to_Common_TSs REST-2010-A103, OMA-ARC-REST-2010-0517R02- CR_Presence_TS_small_corrections OMA-ARC-REST-2010-0483R01- CR_ALM_adding_subscription_and_notification OMA-ARC-REST-2010-0484R01-CR_ALM_adding_shared_contact_lists OMA-ARC-REST-2010-0509R01- CR_ALM_adding_filters_for_shallow_deep_views OMA-ARC-REST-2010-0518-CR_ALM_TS_small_corrections OMA-ARC-REST-2010-0519R03- CR_ALM_changes_to_merged_contact_list Editorial cleanup and validation towards OMA-SUP- XSD_rest_addresslistmgt-V1_0-20100930-D.xsd Generated new JSON
	06 Oct 2010	All	Editorial fixes: Styles Figures caption History table
	07 Oct 2010	5.20.4.2	Fixed paragraph
	07 Nov 2010	All Tables	Merged CR's 587, 588, 599 and 601, closing CONRR comments G0022, G0032, G0041 and G0068 G001/G002 G003, G004, G005, G006
	10 Nov 2010	5.0 5.1 5.1 5.2.x, 5.3.2 5.3.x, 5.4 5.4.3.1.x 5.16.3.1.1	G008 G009 (G010, no change) G011, G0012, G0013 G0023, G0024, G0025, G0030, G0033, G0036, G0037 G0038, G0039, G0040, G0042, G0043, G0046, G0048 G0049, G0051, G0053, G0054, G0055, G0056, G0057 G0058, G0062, G0063, G0064, G0065, G0067
	15 Nov 2010	5.6.1, 5.7.1, 5.11.1,5.22.1 5.x	G0052 solved by OMA-ARC-REST-2010-0616- CR_CONRR_comment_G0052_ALM G0021, G0051, G0069 and G0071 solved by OMA-ARC-REST-2010-0654- CR_CONRR_comment_G0021_G0051_G0069_G0071_ALM
	6 Dec 2010	All 5.2.6	OMA-ARC-REST-2010- 0681R01- CR_Changing_defining_to_filter_in_ALM G007, G015, G016, G018, G026, G027, G028, G029, F044, G045, G059, G060, and G061, from OMA-ARC-REST-2010-0652R02- CR_CONRR_comments_resolution_ALM G0028, OMA-ARC-REST-2010-0712- CR_CONRR_G0028_related_changes_for_ALM
	7 Dec 2010	5.3 5.2.x SCR	OMA-ARC-REST-2010- 0714R01- CR_Sequence_flows_for_authorization_rules_in_ALM OMA-ARC-REST-2010-0716-CR_MinorUpdatesALM OMA-ARC-REST-2010-0704R01- CR_A225_CONRR_R001_resolution_for_ALM OMA-ARC-REST-2010- 0720-CR_SCR_updates_in_ALM All XML examples are valid with ALM XSD version 20101207
	8 Dec 2010		Regenerated JSON examples, part 1 More JSON, part 2

Document Identifier	Date	Sections	Description
	9 Dec 2010	6 5, App F Many	OMA-ARC-REST-2010-0722-CR_ALM_New_Service_Policy_Exceptions OMA-ARC-REST-2010-0723- CR_ALM_New_Appendix_for_lightweight_resources Proofreading and minor corrections Editorials
	14 Dec 2010	D30, D31	Added missing examples, and validated
	16 Dec 2010	Various	Editorial fixes: styles, tables aligned, history table
Candidate Version: OMA-TS- ParlayREST_AddressListManage ment-V1_0	11 Jan 2011	All	Status changed to Candidate by TP: OMA-TP-2010-0531R01-INP_ParlayREST_2_0_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for ParlayREST.ALM Server

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

B.2 SCR for ParlayREST.ALM.ContactLists Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-CONTACT-COL-S-001-O	Support contact collection with all of user's contacts	5.4	PARLAYREST-ALM-CONTACT-COL-S-002-O
PARLAYREST-ALM-CONTACT-COL-S-002-O	This operation retrieves all contacts in the contact collection - GET	5.4.3	

B.3 SCR for ParlayREST.ALM.IndividualContactList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-CONTACT-IND--S-001-O	Support individual contact in contact collection	5.5	PARLAYREST-ALM-CONTACT-IND-S-002-O AND PARLAYREST-ALM-CONTACT-IND-S-003-O AND PARLAYREST-ALM-CONTACT-IND-S-004-O
PARLAYREST-ALM-CONTACT-IND-S-002-O	This operation retrieves information of a contact in the contact collection - GET	5.5.3	
PARLAYREST-ALM-CONTACT-IND-S-003-O	This operation adds or updates a contact in the contact collection - PUT	5.5.4	
PARLAYREST-ALM-CONTACT-IND-S-004-O	This operation deletes a contact in the contact collection - DELETE	5.5.6	

B.3.1 SCR for ParlayREST.ALM.AttributesForAContact Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-ATTRIB-CONTACT-S-001-O	Supports retrieval of all attributes of a contact	5.6	PARLAYREST-ALM-ATTRIB-CONTACT-S-002-O PARLAYREST-ALM-ATTRIB-CONTACT-S-003-O
PARLAYREST-ALM-CONTACT-S-002-O	This operation retrieves all attributes for a contact – GET	5.6.3	

B.3.2 SCR for ParlayREST.ALM.IndividualAttributeForAContact Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-001-O	Support for management (create, update, retrieve and delete) of a contact's single attribute.	5.7	PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-002-O AND PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-003-O AND PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-004-O
PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-002-O	This operation returns the value of the attribute for a member – GET	5.7.3	
PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-003-O	This operation creates or updates an attribute for a member – PUT	5.7.4	
PARLAYREST-ALM-IND-ATTRIB-CONTACT-S-004-O	This operation removes an attribute for a member - DELETE	5.7.6	

B.3.3 SCR for ParlayREST.ALM.MemberLists Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-MEM-LS-S-001-M	Support for retrieving all member lists belonging to a user.	5.8	
PARLAYREST-ALM-MEM-LS-S-002-M	This operation retrieves all member lists the user has created - GET	5.8.3	

B.3.4 SCR for ParlayREST.ALM.IndividualMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-MEM-LS-001-M	Support for management (create, update, retrieve and delete) of an individual member list.	5.9	

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-MEM-LS-002-M	This operation retrieves the member list - GET	5.9.3	
PARLAYREST-ALM-IND-MEM-LS-003-M	This operation creates or updates the entire member list - PUT	5.9.4	
PARLAYREST-ALM-IND-MEM-LS-004-M	This operation removes the member list including attributes and members - DELETE	5.9.6	

B.3.5 SCR for ParlayREST.ALM.AttributesForAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-ATTRIB-MEM-LS-S-001-O	Support for retrieving all attributes for a given member list belonging to a user.	5.10	PARLAYREST-ALM-ATTRIB-MEM-LS-S-002-O
PARLAYREST-ALM-ATTRIB-MEM-LS-S-002-O	This operation returns all attributes for a member list - GET	5.10.3	

B.3.6 SCR for ParlayREST.ALM.IndividualAttributeForAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-001-O	Support for management (create, update, retrieve and delete) of individual attributes for a member list.	5.11	PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-002-O AND PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-003-O AND PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-004-O
PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-002-O	This operation returns the value of the member list attribute - GET	5.11.3	
PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-003-O	This operation creates or updates an attribute - PUT	5.11.4	
PARLAYREST-ALM-IND-ATTRIB-MEM-LS- MGT-004-O	This operation deletes an attribute - DELETE	5.11.6	

B.3.7 SCR for ParlayREST.ALM.MemberInAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-MBR-MEM-LS-S-001-O	Supports retrieval of all members of a member list belonging to a user	5.12	PARLAYREST-ALM-MBR-MEM-LS-S-002-O

Item	Function	Reference	Requirement
PARLAYREST-ALM-MBR-MEM-LS-S-002-O	This operation retrieves the members in the member list - GET	5.12.3	

B.3.8 SCR for ParlayREST.ALM.IndividualMemberInAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-MBR-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of members on individual member list.	5.13	PARLAYREST-ALM-IND-MBR-MEM-LS-S-002-O PARLAYREST-ALM-IND-MBR-MEM-LS-S-003-O PARLAYREST-ALM-IND-MBR-MEM-LS-S-004-O
PARLAYREST-ALM-IND-MBR-MEM-LS-S-002-O	This operation retrieves a user from a member list - GET	5.13.3	
PARLAYREST-ALM-IND-MBR-MEM-LS-S-003-O	This operation creates and updates an entry in the member list - PUT	5.13.4	
PARLAYREST-ALM-IND-MBR-MEM-LS-S-004-O	This operation removes the member from the member list - DELETE	5.13.6	

B.3.9 SCR for ParlayREST.ALM.AttributesForAMemberInAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-ATTRIB-MBR-MEM-LS-S-001-O	Supports retrieval of all attributes of a member of a member list belonging to a user	5.14	PARLAYREST-ALM-ATTRIB-MBR-MEM-LS-S-002-O
PARLAYREST-ALM-ATTRIB-MBR-MEM-LS-S-002-O	This operation returns all attributes for a member – GET	5.14.3	

B.3.10 SCR for ParlayREST.ALM.IndividualAttributeForAMemberInAMemberList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of member attributes on individual	5.15	PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-002-O AND PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-003-O AND PARLAYREST-ALM-IND-

Item	Function	Reference	Requirement
	member list.		ATTRIB-MEM-LS-S-004-O
PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-002-O	This operation returns the value of the attribute for a member – GET	5.15.3	
PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-003-O	This operation creates or updates an attribute for a member – PUT	5.15.4	
PARLAYREST-ALM-IND-ATTRIB-MEM-LS-S-004-O	This operation removes an attribute for a member - DELETE	5.15.6	

B.3.11 SCR for ParlayREST.ALM.MemberListReferences Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested member lists.	5.16	PARLAYREST-ALM-MEM-LS-REF-S-002-O
PARLAYREST-ALM-MEM-LS-REF-S-002-O	This operation returns a list of references to other member lists. – GET	5.16.3	

B.3.12 SCR for ParlayREST.ALM.IndivMemberListReference Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested member lists.	5.17	PARLAYREST-ALM-IND-MEM-LS-REF-S-002-O AND PARLAYREST-ALM-IND-MEM-LS-REF-S-003-O AND PARLAYREST-ALM-IND-MEM-LS-REF-S-004-O
PARLAYREST-ALM-IND-MEM-LS-REF-S-002-O	This operation returns an individual member list reference to another member list – GET	5.17.3	
PARLAYREST-ALM-IND-MEM-LS-REF-S-003-O	This operation creates or updates an individual reference to another member list. – PUT	5.17.4	
PARLAYREST-ALM-IND-MEM-LS-REF-S-004-O	This operation deletes an individual reference to another member list. – DELETE	5.17.6	

B.3.13 SCR for ParlayREST.ALM.List.Subscr Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-SUBSCR-S-001-O	Support subscriptions for list changes notifications	5.18	PARLAYREST-ALM-SUBSCR-S-002-O AND PARLAYREST-ALM-SUBSCR-S-003-O
PARLAYREST-ALM-SUBSCR-S-002-O	This operation retrieves a list of subscriptions - GET	5.18.3	
PARLAYREST-ALM-SUBSCR-S-003-O	This operation creates a subscription for list or contact collection change notifications - POST	5.18.5 C.1	

B.3.14 SCR for ParlayREST.ALM.Individual.Subscr Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-SUBSCR-S-001-O	Support for managing an individual subscription for list or contact collection change notifications	5.19	PARLAYREST-ALM-IND-SUBSCR-S-002-O AND PARLAYREST-ALM-IND-SUBSCR-S-003-O AND PARLAYREST-ALM-IND-SUBSCR-S-004-O
PARLAYREST-ALM-IND-SUBSCR-S-002-O	This operation retrieves an individual subscription for list or contact collection change notifications - GET	5.19.3	
PARLAYREST-ALM-IND-SUBSCR-S-003-O	This operation updates an individual subscription for list or contact collection change notifications - PUT	5.19.4	
PARLAYREST-ALM-IND-SUBSCR-S-004-O	This operation deletes an individual subscription for list or contact collection change notifications - DELETE	5.19.6	

B.3.15 SCR for ParlayREST.ALM.Notif Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-NOTIF-S-001-O	Support for notifying application about list or contact collection changes	5.20	PARLAYREST-ALM-NOTIF-S-002-O
PARLAYREST-ALM-NOTIF-S-002-O	Notify application about list or contact collection changes - POST	5.20.5	

B.3.16 SCR for ParlayREST.ALM.Shared.ContactCollection Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-SHAREDCONCOLL-S-001-O	Support shared contact collection	5.21	PARLAYREST-ALM-SHAREDCONCOLL-S-002-O
PARLAYREST-ALM-SHAREDCONCOLL-S-002-O	This operation the collection of shared contacts by one user with another user - GET	5.21.3	

B.3.17 SCR for ParlayREST.ALM.Individual.Shared.Contact Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-SHAREDCON-S-001-O	Support individual shared contact	5.22	PARLAYREST-ALM-IND-SHAREDCON-S-002-O
PARLAYREST-ALM-IND-SHAREDCON-S-002-O	This operation retrieves an individual shared contact by one user with another user - GET	5.22.3	

B.3.18 SCR for ParlayREST.ALM.Shared.Lists Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-SHAREDLISTS-S-001-O	Support shared member lists	5.23	PARLAYREST-ALM-SHAREDLISTS-S-002-O
PARLAYREST-ALM-SHAREDLISTS-S-002-O	This operation retrieves a collection of all shared member lists by one user with another user - GET	5.23.3	

B.3.19 SCR for ParlayREST.ALM.Individual.Shared.List Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-SHAREDLIST-S-001-O	Support individual shared member list	5.24	PARLAYREST-ALM-IND-SHAREDLIST-S-002-O
PARLAYREST-ALM-IND-SHAREDLIST-S-002-O	This operation retrieves an individual shared member list by one user with another user - GET	5.24.3	

B.3.20 SCR for ParlayREST.ALM.Member.Shared.List Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-MEMBER-SHAREDLIST-S-001-O	Support individual member in shared member list	5.25	PARLAYREST-ALM-MEMBER-SHAREDLIST-S-002-O
PARLAYREST-ALM-MEMBER-SHAREDLIST-S-002-O	This operation retrieves information of an individual member in a shared member list - GET	5.25.3	

B.3.21 SCR for ParlayREST.ALM.Authorization.RuleList Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-AUTH-RULE-LS-S-001-O	Support authorization rules for sharing member lists or contacts	5.26	PARLAYREST-ALM-AUTH-RULE-LS-S-002-O AND PARLAYREST-ALM-AUTH-RULE-LS-S-003-O
PARLAYREST-ALM-AUTH-RULE-LS-S-002-O	This operation retrieves all authorization rules owned by a given user - GET	5.26.3	
PARLAYREST-ALM-AUTH-RULE-LS-S-003-O	This operation creates a new authorization rule - POST	5.26.5 C.2	

B.3.22 SCR for ParlayREST.ALM.Indiv.AuthorizationRule Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-AUTH-RULE-S-001-O	Support individual authorization rule	5.27	PARLAYREST-ALM-IND-AUTH-RULE-S-002-O AND PARLAYREST-ALM-IND-AUTH-RULE-S-003-O AND PARLAYREST-ALM-IND-AUTH-RULE-S-004-O
PARLAYREST-ALM-IND-AUTH-RULE-S-002-O	This operation retrieves an individual authorization rule - GET	5.27.3	
PARLAYREST-ALM-IND-AUTH-RULE-S-003-O	This operation updates an individual authorization rule - PUT	5.27.4	
PARLAYREST-ALM-IND-AUTH-RULE-S-004-O	This operation deletes an individual authorization rule - DELETE	5.27.6	

B.3.23 SCR for ParlayREST.ALM.Indiv.AuthorizationRuleData Server

Item	Function	Reference	Requirement
PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-001-O	Support individual authorization rule data	5.28	PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-002-O AND PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-003-O AND PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-004-O
PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-002-O	This operation retrieves an individual authorization rule specific data - GET	5.28.3	
PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-003-O	This operation updates an individual authorization rule specific data - PUT	5.28.4	
PARLAYREST-ALM-IND-AUTH-RULE-DATA-S-004-O	This operation deletes an individual authorization rule specific data - DELETE	5.28.6	

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

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

The encoding is defined below for all AddressListManagement REST operations which are based on POST requests.

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.

The following AddressListManagement REST operations are defined in this section:

- Create a subscription for member list changes notification
- Create an authorization rule

C.1 Create a subscription for member list changes notifications

This operation is used to create a new subscription for member list changes notifications.

This REST operation is used by the application to start the member list changes notifications. 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
anyContacts	xsd:boolean	Choice	Indicates a subscription for changes about any contacts in the ContactCollection.
memberListId	xsd:anyURI	Choice	Indicates a subscription for changes about the specified member list.
notifyURL	xsd:anyURI	No	Notification endpoint definition
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	<p>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</p> <p>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.</p> <p>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.</p>
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	<p>Default: XML</p> <p>Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}</p>

C.1.1 Example

(Informative)

C.1.1.1 Request

```
POST /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/subscriptions HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Accept: application/xml
```

```
anyContacts=true&
notifyURL=http://client.example.com/notifications/memberLists&
callbackData=12345&
clientCorrelator=123&
applicationTag=myApp&
duration=3600&
notificationFormat=XML
```

C.1.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 21:33:52 GMT
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321

```
<?xml version="1.0" encoding="UTF-8"?>
<alm:listChangesSubscription xmlns:alm="urn:oma:xml:rest:addresslistmgt:1">
  <anyContacts/>
  <callbackReference>
    <notifyURL>http://client.example.com/notifications/contacts</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>123</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <duration>3600</duration>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-
0100/subscriptions/listChanges/987654321</resourceURL>
</alm:listChangesSubscription>
```

C.2 Create an authorization rule

This operation is used to create an authorization rule. 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. It is defined as an attribute in XML format
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 specified contact identity in the ContactCollection.
applToAllContacts	xsd:boolean	Choice1	The rule applies to all contacts.
applToMemLists	xsd:anyURI [0..unbounded]	Choice2	The rule applies to the specified member list.
applToAllMemLists	xsd:boolean	Choice2	The rule applies to all member lists.
applToMemListsAndContacts	xsd:anyURI [0..unbounded]	Choice3	The rule applies to the specified member list. The rule applies also to contacts that are referenced from the specified member list.
applToAllMemListsAndContacts	xsd:boolean	Choice3	The rule applies for all member lists and linked 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.
authorizedToMemberListId	xsd:anyURI [0..unbounded]	Choice4	Users within the member list are authorized to access a resource for which the rule is applicable to.
authorizedToAllMemberLists	xsd:boolean	Choice4	Users within all member lists are authorized to access a resource for which the rule is applicable to.
authorizedToExternalMemberList	xsd:anyURI [0..unbounded]	Choice4	Users within the member 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 member list attributes (according to 5.4) 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 (according to 5.4) the retrieving user is allowed to see. An empty filter means that the user has access to all attributes.

C.2.1 Example

(Informative)

C.2.1.1 Request

```
POST /exampleAPI/1/addresslistmgt/ tel%3A%2B1-555-555-0100/memberLists/subscriptions HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Accept: application/xml

ruleName=combinedRule&
clientCorrelator=135&
applicationTag=myApp&
applToMemListsAndContacts=5678&
authorizedToMemberListId=5678
```

C.2.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 21:33:52 GMT
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3

<?xml version="1.0" encoding="UTF-8"?>
<alm:rule xmlns:alm="urn:oma:xml:rest:addresslistmgt:1" ruleName="combinedRule">
  <clientCorrelator>135</clientCorrelator>
  <applicationTag>myApp</applicationTag>
  <applToMemListsAndContacts>
    <memberListId>5678</memberListId>
  </applToMemListsAndContacts>
  <authorizedTo>
    <memberListId>5678</memberListId>
  </authorizedTo>
  <resourceURL>http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3</resourceURL></alm:rule>
```

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 Parlay REST 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 [OMA_REST_TS_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [OMA_REST_TS_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 5.4.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts HTTP/1.1
```

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

Response:

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

```
{  
  "contactCollection": {  
    "contact": [  
      {  
        "attributeList": {  
          "attribute": [  
            {  
              "name": "display-name",  
              "value": "Alice"  
            },  
            {  
              "name": "cellphone",  
              "value": "tel:+1-555-555-0200"  
            },  
            {  
              "name": "state",  
              "value": "California"  
            }  
          ]  
        }  
      }  
    ]  
  },  
  "contactId": "alice",  
  "link": {  
    "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122",  
    "rel": "Member"  
  }  
}
```

```

    },
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice"
  },
  {
    "attributeList": {"attribute": [
      {
        "name": "display-name",
        "value": "Sam"
      },
      {
        "name": "cellphone",
        "value": "tel:+1-555-555-0300"
      },
      {
        "name": "state",
        "value": "New Jersey"
      }
    ]},
    "contactId": "sam",
    "link": [
      {
        "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0121",
        "rel": "Member"
      },
      {
        "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678/members/tel%3A%2B1-555-555-0121",
        "rel": "Member"
      }
    ],
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/sam",
    "sharedIdentity": {"sharedId": "tel:+1-555-555-0121"}
  }
],
"resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts"
}}

```

D.2 Retrieve individual contact with all attributes (default) (section 5.5.3.1)

Request:

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK

```

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

{"contact": {
  "attributeList": {"attribute": [
    {
      "name": "display-name",
      "value": "Alice"
    },
    {
      "name": "cellphone",
      "value": "tel:+1-555-555-0200"
    },
    {
      "name": "state",
      "value": "California"
    }
  ]},
  "contactId": "alice",
  "link": {
    "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122",
    "rel": "Member"
  },
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice"
}}
```

D.3 Get all attributes of a contact (section 5.6.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria/attributes HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"attributeList": {
  "attribute": {
    "name": "cellphone",
    "value": "tel:+1-555-555-0255"
  },
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/maria/attributes"
}}
```


D.4 Retrieve a contact's attribute (section 5.7.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/cellphone HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

```
{"attribute": {
  "name": "cellphone",
  "value": "tel:+1-555-555-0200"
}}
```

D.5 Create or update a contact's attribute (section 5.7.4.1)

Request:

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/alice/attributes/married HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: example.com
{"attribute": {
  "name": "married",
  "value": "true"
}}
```

Response:

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

```
{"attribute": {
  "name": "married",
  "value": "true"
}}
```

D.6 Retrieve specific member lists attributes and specific member attributes (section 5.8.3.3)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists?listFilter=display-name&indivFilter=display-name&indivFilter=phone HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"memberListCollection": {"memberList": [
  {
    "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Friends"
    }},
    "category": "URList",
    "memberCollection": {"member": {
      "attributeList": {"attribute": {
        "name": "display-name",
        "value": "Alice"
      }},
      "memberId": "sip:alice@example.com",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com"
    }},
    "memberListId": "1234",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234",
    "sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
  },
  {
    "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Family"
    }},
    "memberCollection": {"member": [
      {
        "attributeList": {"attribute": {
          "name": "display-name",
          "value": "Wife"
        }},
        "memberId": "mailto:liza@example.com",
        "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aliza@example.com"
      }
    ]}
  }
]}
}
```

```

    },
    {
      "attributeList": {"attribute": [
        {
          "name": "display-name",
          "value": "Son"
        },
        {
          "name": "phone",
          "value": "tel:+1-555-555-0260"
        }
      ]},
      "memberId": "mailto:serezha@example.com",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/members/mailto%3Aserezha@example.com"
    }
  ]},
  "memberListId": "5678",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678"
}
]]}

```

D.7 Retrieve a member list with all attributes and all member attributes (default) (section 5.9.3.1)

Request:

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234 HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

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

{"memberList": {
  "attributeList": {"attribute": {
    "name": "display-name",
    "value": "Friends"
  }},
  "memberCollection": {"member": {
    "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Alice"
    }},
    "memberId": "sip:alice@example.com",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-

```

```
0100/memberLists/1234/members/sip:alice@example.com"
  }},
  "memberListId": "1234",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234"
}}
```

D.8 Retrieve a non existing member list (section 5.9.3.4)

Request:

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/4321 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"requestError": {
  "link": {
    "href": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/4321",
    "rel": "MemberList"
  },
  "serviceException": {
    "messageId": "SVC0002",
    "text": "Invalid input value for message part %1",
    "variables": "memberListId"
  }
}}
```

D.9 Get all attributes for a member lists belonging to Bob (section 5.10.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"attributeList": {
  "attribute": [
    {
      "name": "display-name",
      "value": "Friends"
    },
    {
      "name": "date-created",
      "value": "Fri, 10 Dec 2010 11:28:34 GMT"
    }
  ]
},
"resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes"
}}
```

D.10 Retrieve an attribute value (section 5.11.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/date-created HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"attribute": {
  "name": "date-created",
  "value": "Fri, 10 Dec 2010 11:28:34 GMT"
}}
```

D.11 Create an attribute(section 5.11.4.1)

Request:

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/attributes/display-name HTTP/1.1
Accept: application/json
Host: example.com
```

```
{
  "attribute": {
    "name": "display-name",
    "value": "Close friends"
  }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Location:http://{serverRoot}/{apiVersion}/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/{memberListId}
Date: Fri, 10 Dec 2010 12:51:59 GMT
```

```
{
  "attribute": {
    "name": "display-name",
    "value": "Close friends"
  }
}
```

D.12 Get all members for a member list belonging to a user with all attributes (default) (section 5.12.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT
Fri, 10 Dec 2010 {"memberCollection": {
  "member": [
    {
      "attributeList": {"attribute": {
        "name": "display-name",
        "value": "Alice"
      }},
      "memberId": "mailto:alice@example.com",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com"
    },
    {
      "attributeList": {"attribute": {
        "name": "display-name",
```

```

    "value": "Bob"
  },
  "memberId": "tel:+1-555-555-0124",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0124"
}
],
"resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members"
}}

```

D.13 Retrieve a member and selected attributes (section 5.13.3.2)

Request:

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com?indivFilter=cellphone Accept: application/json
Host: example.com

```

Response:

```

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

{"member": {
  "attributeList": {"attribute": {
    "name": "cellphone",
    "value": "tel:+1-555-555-0155"
  }},
  "memberId": "sip:alice@example.com",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com"
}}

```

D.14 Get all attributes of a member of a member list (section 5.14.3.1)

Request:

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/mailto%3Aalice@example.com/attributes
HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

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

{"attributeList": {
  "attribute": [
    {
      "name": "display-name",
      "value": "Alice"
    },
    {
      "name": "cellphone",
      "value": "tel:+1-555-555-0155"
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234/members/mailto%3Aalice@example.com/attributes"
}}
```

D.15 Retrieve a non existing attribute (section 5.15.3.2)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/sip:alice@example.com/attributes/pet
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:51:59 GMT

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

D.16 Create a member's attribute (section 5.15.4.1)

Request:


```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/sip:alice@example.com/attributes/born
HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: example.com

{"attribute": {
  "name": "born",
  "value": "1984"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/6789/members/sip:alice@example.com/attributes/born
Date: Fri, 10 Dec 2010 12:51:59 GMT

{"attribute": {
  "name": "born",
  "value": "1984"
}}
```

D.17 Delete an attribute (section 5.16.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/allFriends/memberListReferences HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"memberListReferenceCollection": {
  "link": [
    {
      "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/1234",
      "rel": "MemberList"
    },
    {
      "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678",
      "rel": "MemberList"
    }
  ]
}}
```

```
  ],  
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-  
0100/memberLists/friends/memberListReferences"  
}}
```

D.18 Create a member list reference (section 5.17.4.1)

Request:

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-  
0100/memberlist/allFriends/memberListReferences/http%3A//example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-  
0100/memberLists/5678 HTTP/1.1  
Content-Type: application/xml  
Accept: application/json  
Host: example.com  
  
{"link": {  
  "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678",  
  "rel": "MemberList"  
}}
```

Response:

```
HTTP/1.1 201 Created  
Content-Type: application/json  
Location: http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberlist/allFriends/memberListReferences/  
http%3A//example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678  
Date: Fri, 10 Dec 2010 12:51:59 GMT  
  
{"link": {  
  "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/memberLists/5678",  
  "rel": "MemberList"  
}}
```

D.19 Get member list changes subscriptions (section 5.18.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges HTTP/1.1  
Accept: application/json  
Host: example.com
```

Response:

```

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

{"listChangesSubscriptionCollection": {
  "listChangesSubscription": [
    {
      "applicationTag": "myApp",
      "callbackReference": {
        "callbackData": "12345",
        "notifyURL": "http://client.example.com/notifications/memberLists/1234"
      },
      "clientCorrelator": "123",
      "duration": "1958",
      "memberListId": "1234",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321"
    },
    {
      "anyContacts": null,
      "applicationTag": "myApp",
      "callbackReference": {
        "callbackData": "54321",
        "notifyURL": "http://client.example.com/notifications/contacts"
      },
      "clientCorrelator": "456",
      "duration": "2641",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/123456789"
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges"
}}

```

D.20 Get member list changes subscription (section 5.19.3.1)

Request:

```

GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321 HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

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

```

```
{
  "listChangesSubscription": {
    "applicationTag": "myApp",
    "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://client.example.com/notifications/memberLists/1234"
    },
    "clientCorrelator": "123",
    "duration": "6817",
    "memberListId": "1234",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/987654321"
  }
}
```

D.21 Notification for contact change subscription (section 5.20.5.2)

Request:

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

```
{
  "listChangeNotification": {
    "callbackData": "54321",
    "duration": "4217",
    "link": [
      {
        "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/contacts/ole",
        "rel": "Contact"
      },
      {
        "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/subscriptions/listChanges/123456789",
        "rel": "ListChangesSubscription"
      }
    ],
    "resourceStatus": "Active"
  }
}
```

Response:

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

D.22 Get contacts shared by a user identified by otherUserId (section 5.21.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"contactCollection": {
  "contact": [
    {
      "attributeList": {"attribute": {
        "name": "display-name",
        "value": "Alice"
      }},
      "contactId": "alice",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/sip:alice@example.com",
      "sharedIdentity": {"sharedId": "sip:alice@example.com"}
    },
    {
      "attributeList": {"attribute": {
        "name": "display-name",
        "value": "Ole"
      }},
      "contactId": "ole",
      "link": {
        "href": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/shared/memberLists/6789/members/mailto%3Aole@example.com",
        "rel": "Member"
      },
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/mailto%3Aole@example.com",
      "sharedIdentity": {"sharedId": "mailto:ole@example.com"}
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts"
}}
```

D.23 Create a member without a link to the Member (section 5.22.3.2)

Request:

```
GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/contacts/alice HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"contact": {
  "attributeList": {"attribute": [
    {
      "name": "display-name",
      "value": "Alice"
    },
    {
      "name": "cellphone",
      "value": "tel:+1-555-555-0200"
    },
    {
      "name": "state",
      "value": "California"
    }
  ]},
  "contactId": "alice",
  "link": {
    "href": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/tel%3A%2B1-555-555-0122",
    "rel": "Member"
  },
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/alice"
}}
```

D.24 Get member lists shared by a user identified by otherUserId without member lists attributes, but with selected member attributes (section 5.23.3.2)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists?listFilter=~noAttr&
indivFilter=phone HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"memberListCollection": {"memberList": [
  {
    "category": "URIList",
    "memberCollection": {"member": {
      "memberId": "sip:alice@example.com",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com"
    }},
    "memberListId": "1234",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234",
    "sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
  },
  {
    "memberCollection": {"member": [
      {
        "memberId": "mailto:liza@example.com",
        "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678/members/mailto%3Aliza@example.com"
      },
      {
        "attributeList": {"attribute": {
          "name": "phone",
          "value": "tel:+1-555-555-0260"
        }},
        "memberId": "mailto:serezha@example.com",
        "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678/members/mailto%3Aserezha@example.com"
      }
    ]},
    "memberListId": "5678",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/5678"
  }
]}
}}
```

D.25 Create a new member of the member list (section 5.24.3.1)**Request:**

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"memberList": {
  "attributeList": {"attribute": {
    "name": "display-name",
    "value": "Friends"
  }},
  "category": "URList",
  "memberCollection": {"member": {
    "attributeList": {"attribute": {
      "name": "display-name",
      "value": "Alice"
    }},
    "memberId": "sip:alice@example.com",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip:alice@example.com"
  }},
  "memberListId": "1234",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234",
  "sharedListIdentity": {"sharedId": "sip:bob@example.com;list=1234"}
}}
```

D.26 Get individual member information about user identified as memberId from the list shared by a user identified by otherUserId (section 5.25.3.2)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip%3Aalice@example.com?indivFilter=cellphone HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
```



```

Content-Type: application/json
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:53:23 GMT

{"member": {
  "attributeList": {"attribute": {
    "name": "cellphone",
    "value": "tel:+1-555-555-0105"
  }},
  "memberId": "sip:alice@example.com",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0260/shared/tel%3A%2B1-555-555-0100/memberLists/1234/members/sip%3Aalice@example.com"
}}

```

D.27 Get authorization rules (section 5.26.3.1)

Request:

```

GET /exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/authorizationRules HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

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

{"ruleList": {
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/rules",
  "rule": [
    {
      "appToContacts": {"allContacts": null},
      "applicationTag": "myApp",
      "authorizedTo": {"memberListId": "1234"},
      "clientCorrelator": "123",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/rules/rule1",
      "ruleName": "allowAllContacts"
    },
    {
      "appToContacts": {"contactId": "bob"},
      "applicationTag": "myApp",
      "authorizedTo": {"allSharedIdentities": null},
      "clientCorrelator": "456",
      "resourceURL": "http://example.com/exampleAPI/1/addresslistmgmt/tel%3A%2B1-555-555-0100/rules/rule2",
      "ruleName": "allowOwnContact"
    }
  ],
  "appToMemLists": {"memberListId": "1234"},
}

```

```

    "applicationTag": "myApp",
    "authorizedTo": {"memberListId": "1234"},
    "clientCorrelator": "789",
    "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule3",
    "ruleName": "allowFriendsList"
  }
]
}}

```

D.28 Update an existing authorization rule (section 5.27.4.1)

Request:

```

PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule1 HTTP/1.1
Content-Type: application/xml
Content-Length : nnnn
Accept: application/json
Host: example.com

{"rule": {
  "appToContacts": {"allContacts": null},
  "applicationTag": "myApp",
  "authorizedTo": {"memberListId": [
    "1234",
    "5678"
  ]},
  "clientCorrelator": "123",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1",
  "ruleName": "allowAllContacts"
}}

```

Response:

```

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

{"rule": {
  "appToContacts": {"allContacts": null},
  "applicationTag": "myApp",
  "authorizedTo": {"memberListId": [
    "1234",
    "5678"
  ]},
  "clientCorrelator": "123",
  "resourceURL": "http://example.com/exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/rules/rule1",
  "ruleName": "allowAllContacts"
}}

```

D.29 Retrieve data from an authorization rule (section 5.28.3.1)

Request:

```
GET /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule1/memberLists/1234 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:55:59 GMT
```

```
{"memberListId": "1234"}
```

D.30 Authorize user by updating an existing authorization rule (section 5.28.4.1)

Request:

```
PUT /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2/users/mailto%3Aalice@example.com HTTP/1.1
Content-Type: application/xml
Content-Length : nnnn
Accept: application/json
Host: example.com
```

```
{"userId": "mailto:alice@example.com"}
```

Response:

```
HTTP/1.1 201 Created
Location: /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/authorizationRules/rule2/users/mailto%3Aalice@example.com
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 10 Dec 2010 12:55:59 GMT
```

```
{"userId": "mailto:alice@example.com"}
```

D.31 Delete an authorized identity (section 5.28.6.1)

Request:

```
DELETE /exampleAPI/1/addresslistmgt/tel%3A%2B1-555-555-0100/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
```

Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/methods and Parlay X equivalent operations.

ParlayREST Resource	ParlayREST Method	ParlayREST Section reference	Parlay X equivalent operation
ContactLists	GET	5.8.3	getOwnersGroups
Individual member list	DELETE	5.9.6	deleteGroup
Attributes for a member list	GET	5.10.3	queryGroupAttributes
Individual attribute for a member list	DELETE	5.11.6	deleteGroupAttribute
Members in a member list	GET	5.12.3	queryMembers
Individual member in a member list	DELETE	5.13.6	deleteMember
Attributes for a member in a member list	GET	5.14.3	queryGroupMemberDefiner
Individual attribute for a member in a member list	DELETE	5.15.6	deleteGroupMemberAttribute

Table 3: Parlay X operations mapping

Appendix F. Light-weight resources for Address List Management (Informative)

The following table lists all Address List Management 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 (5.2.6)	attribute	attribute	Attribute	{name}
Member list references (5.2.10)	link	link	common:Link	{href}
Authorization (5.2.19)	userId	userId	xsd:anyURI	users/{userId}
	memberListId	memberListId	xsd:anyURI	memberLists/{memberListId}
	link	link	common:Link	externalMemberLists/{href}
	domainName	domainName	xsd:string	domains/{domainName}

Table 4: Light-weight resources for Address List Management

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. Overview of data types (Informative)

The section provides an overview of the relation between the data types used for contacts, members and member lists.

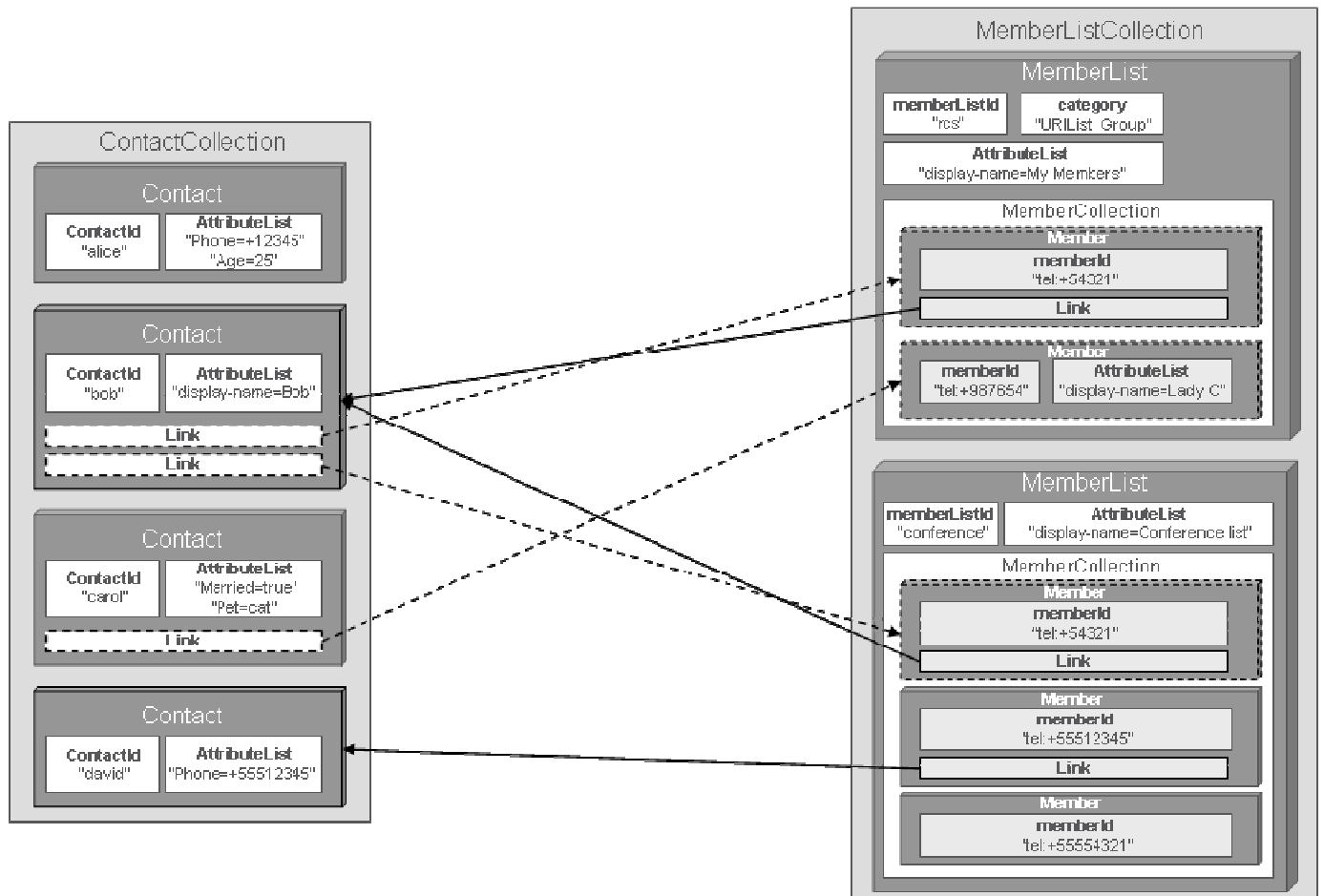


Figure 6 Data types for contacts, members and member lists