



RESTful bindings for Parlay X Web Services - Payment

Candidate Version 1.1 – 11 Jan 2011

Open Mobile Alliance
OMA-TS-ParlayREST_Payment-V1_1-20110111-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.	PAYMENT API DEFINITION	15
5.1	RESOURCES SUMMARY	15
5.2	PAYMENT PARLAYREST API DATA STRUCTURES	20
5.2.1	Type: PaymentTransactionList	21
5.2.2	Type: AmountTransaction	21
5.2.3	Type: AmountSplitTransaction.....	23
5.2.4	Type: VolumeTransaction.....	24
5.2.5	Type: VolumeSplitTransaction.....	26
5.2.6	Type: AmountReservationTransaction	27
5.2.7	Type: VolumeReservationTransaction.....	29
5.2.8	Type: PaymentAmount	31
5.2.9	Type: PaymentVolume	31
5.2.10	Type: ChargingMetaData.....	32
5.2.11	Type: EndUserShare	33
5.2.12	Type: RatingParameter	33
5.2.13	Enumeration: TransactionOperationStatus	33
5.2.14	Values of the Link “rel” attribute.....	34
5.3	SEQUENCE DIAGRAMS.....	34
5.3.1	Amount charge and refund transaction	34
5.3.2	Amount split charge transaction	35
5.3.3	Volume charge and refund transaction	36
5.3.4	Volume split charge transaction.....	36
5.3.5	Amount reservation transaction	37
5.3.6	Volume reservation transaction	38
5.3.7	Amount converted from volume transaction.....	40
5.4	RESOURCE: ALL PAYMENT TRANSACTIONS FOR AN END USER	40
5.4.1	Request URI variables	41
5.4.2	Response Codes	41
5.4.2.1	HTTP Response Codes.....	41
5.4.2.2	Exception fault codes	41
5.4.3	GET.....	41
5.4.3.1	Example 1: get all transactions (Informative).....	41
5.4.3.1.1	Request	41
5.4.3.1.2	Response.....	41
5.4.3.2	Example 2: request with invalid (non-existing) endUserId (Informative).....	45
5.4.3.2.1	Request	45
5.4.3.2.2	Response	46
5.4.4	PUT.....	46
5.4.5	POST.....	46
5.4.6	DELETE	46
5.5	RESOURCE: ALL AMOUNT CHARGE AND REFUND TRANSACTIONS FOR AN END USER	46
5.5.1	Request URI variables	46
5.5.2	Response Codes	47

5.5.2.1	HTTP Response Codes.....	47
5.5.2.2	Exception fault codes.....	47
5.5.3	GET.....	47
5.5.3.1	Example: get all amount transactions (Informative).....	47
5.5.3.1.1	Request.....	47
5.5.3.1.2	Response.....	47
5.5.4	PUT.....	48
5.5.5	POST.....	48
5.5.5.1	Example 1: create charge amount (Informative).....	48
5.5.5.1.1	Request.....	48
5.5.5.1.2	Response.....	49
5.5.5.2	Example 2: create refund amount (Informative).....	49
5.5.5.2.1	Request.....	49
5.5.5.2.2	Response.....	50
5.5.5.3	Example 3: client retries POST with same clientCorrelator (Informative).....	51
5.5.5.3.1	Request.....	51
5.5.5.3.2	Response.....	51
5.5.5.4	Example 4: unsuccessful charge request because of denial/refusal by back-end system (Informative).....	52
5.5.5.4.1	Request.....	52
5.5.5.4.2	Response.....	52
5.5.6	DELETE.....	52
5.6	RESOURCE: ALL AMOUNT SPLIT CHARGE TRANSACTIONS FOR AN END USER.....	53
5.6.1	Request URI variables.....	53
5.6.2	Response Codes.....	53
5.6.2.1	HTTP Response Codes.....	53
5.6.2.2	Exception fault codes.....	53
5.6.3	GET.....	53
5.6.3.1	Example: get all amount split transactions (Informative).....	53
5.6.3.1.1	Request.....	53
5.6.3.1.2	Response.....	53
5.6.4	PUT.....	55
5.6.5	POST.....	55
5.6.5.1	Example: create split charge amount (Informative).....	55
5.6.5.1.1	Request.....	55
5.6.5.1.2	Response.....	55
5.6.6	DELETE.....	56
5.7	RESOURCE: INDIVIDUAL AMOUNT CHARGE OR REFUND TRANSACTION FOR AN END USER.....	56
5.7.1	Request URI variables.....	57
5.7.2	Response Codes.....	57
5.7.2.1	HTTP Response Codes.....	57
5.7.2.2	Exception fault codes.....	57
5.7.3	GET.....	57
5.7.3.1	Example: get amount charge (Informative).....	57
5.7.3.1.1	Request.....	57
5.7.3.1.2	Response.....	57
5.7.4	PUT.....	58
5.7.5	POST.....	58
5.7.6	DELETE.....	58
5.8	RESOURCE: INDIVIDUAL AMOUNT SPLIT CHARGE TRANSACTION FOR AN END USER.....	58
5.8.1	Request URI variables.....	58
5.8.2	Response Codes.....	59
5.8.2.1	HTTP Response Codes.....	59
5.8.2.2	Exception fault codes.....	59
5.8.3	GET.....	59
5.8.3.1	Example: get amount split charge (Informative).....	59
5.8.3.1.1	Request.....	59
5.8.3.1.2	Response.....	59
5.8.4	PUT.....	60
5.8.5	POST.....	60
5.8.6	DELETE.....	60
5.9	RESOURCE: ALL VOLUME CHARGE AND REFUND TRANSACTIONS FOR AN END USER.....	60

5.9.1	Request URI variables	60
5.9.2	Response Codes	60
5.9.2.1	HTTP Response Codes.....	60
5.9.2.2	Exception fault codes.....	60
5.9.3	GET.....	61
5.9.3.1	Example: get all volume charge and refund transactions (Informative).....	61
5.9.3.1.1	Request.....	61
5.9.3.1.2	Response.....	61
5.9.4	PUT.....	62
5.9.5	POST.....	62
5.9.5.1	Example 1: create charge volume, returning a representation of created resource (Informative).....	62
5.9.5.1.1	Request.....	62
5.9.5.1.2	Response.....	62
5.9.5.2	Example 2: create charge volume, returning the location of created resource (Informative).....	63
5.9.5.2.1	Request.....	63
5.9.5.2.2	Response.....	63
5.9.5.3	Example 3: create refund volume (Informative).....	64
5.9.5.3.1	Request.....	64
5.9.5.3.2	Response.....	64
5.9.6	DELETE.....	65
5.10	RESOURCE: ALL VOLUME SPLIT CHARGE TRANSACTIONS FOR AN END USER	65
5.10.1	Request URI variables	65
5.10.2	Response Codes	65
5.10.2.1	HTTP Response Codes.....	65
5.10.2.2	Exception fault codes.....	65
5.10.3	GET.....	66
5.10.3.1	Example: get all volume split charge transactions (Informative).....	66
5.10.3.1.1	Request.....	66
5.10.3.1.2	Response.....	66
5.10.4	PUT.....	67
5.10.5	POST.....	67
5.10.5.1	Example: create volume split charge (Informative).....	67
5.10.5.1.1	Request.....	67
5.10.5.1.2	Response.....	67
5.10.6	DELETE.....	68
5.11	RESOURCE: INDIVIDUAL VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER.....	68
5.11.1	Request URI variables	68
5.11.2	Response Codes	69
5.11.2.1	HTTP Response Codes.....	69
5.11.2.2	Exception fault codes.....	69
5.11.3	GET.....	69
5.11.3.1	Example: get volume charge (Informative).....	69
5.11.3.1.1	Request.....	69
5.11.3.1.2	Response.....	69
5.11.4	PUT.....	70
5.11.5	POST.....	70
5.11.6	DELETE.....	70
5.12	RESOURCE: INDIVIDUAL VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER	70
5.12.1	Request URI variables	70
5.12.2	Response Codes	70
5.12.2.1	HTTP Response Codes.....	70
5.12.2.2	Exception fault codes.....	70
5.12.3	GET.....	71
5.12.3.1	Example: get volume split charge (Informative).....	71
5.12.3.1.1	Request.....	71
5.12.3.1.2	Response.....	71
5.12.4	PUT.....	71
5.12.5	POST.....	72
5.12.6	DELETE.....	72
5.13	RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER.....	72

5.13.1	Request URI variables	72
5.13.2	Response Codes	72
5.13.2.1	HTTP Response Codes	72
5.13.2.2	Exception fault codes.....	72
5.13.3	GET.....	72
5.13.3.1	Example: get amount for volume charge (Informative).....	73
5.13.3.1.1	Request.....	73
5.13.3.1.2	Response.....	73
5.13.4	PUT.....	73
5.13.5	POST.....	73
5.13.6	DELETE	73
5.14	RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER.....	73
5.14.1	Request URI variables	74
5.14.2	Response Codes	74
5.14.2.1	HTTP Response Codes	74
5.14.2.2	Exception fault codes.....	74
5.14.3	GET.....	74
5.14.3.1	Example: get amount for volume split charge (Informative).....	74
5.14.3.1.1	Request.....	74
5.14.3.1.2	Response.....	74
5.14.4	PUT.....	75
5.14.5	POST.....	75
5.14.6	DELETE	75
5.15	RESOURCE: ALL AMOUNT RESERVATION TRANSACTIONS FOR AN END USER.....	75
5.15.1	Request URI variables	75
5.15.2	Response Codes	75
5.15.2.1	HTTP Response Codes	75
5.15.2.2	Exception fault codes.....	75
5.15.3	GET.....	75
5.15.3.1	Example: get all amount reservation transactions (Informative).....	76
5.15.3.1.1	Request.....	76
5.15.3.1.2	Response.....	76
5.15.4	PUT.....	78
5.15.5	POST.....	78
5.15.5.1	Example: create amount reservation (Informative).....	78
5.15.5.1.1	Request.....	78
5.15.5.1.2	Response.....	78
5.15.6	DELETE	79
5.16	RESOURCE: INDIVIDUAL AMOUNT RESERVATION TRANSACTION FOR AN END USER.....	79
5.16.1	Request URI variables	79
5.16.2	Response Codes	79
5.16.2.1	HTTP Response Codes	79
5.16.2.2	Exception fault codes.....	80
5.16.3	GET.....	80
5.16.3.1	Example: get amount reservation (Informative).....	80
5.16.3.1.1	Request.....	80
5.16.3.1.2	Response.....	80
5.16.4	PUT.....	80
5.16.5	POST.....	81
5.16.5.1	Example 1: charge amount for amount reservation (informative).....	81
5.16.5.1.1	Request.....	81
5.16.5.1.2	Response.....	81
5.16.5.2	Example 2: repeat a charge request with same referenceSequence for an amount reservation transaction (Informative) 82	
5.16.5.2.1	Request.....	82
5.16.5.2.2	Response.....	82
5.16.5.3	Example 3: release amount reservation (Informative).....	83
5.16.5.3.1	Request.....	83
5.16.5.3.2	Response.....	83
5.16.5.4	Example 4: charge partial amount for amount reservation (Informative).....	84
5.16.5.4.1	Request.....	84

- 5.16.5.4.2 Response 84
- 5.16.5.5 *Example 5: release remaining amount reservation (Informative)* 85
 - 5.16.5.5.1 Request..... 85
 - 5.16.5.5.2 Response..... 86
- 5.16.5.6 *Example 6: reserve additional amount for amount reservation (Informative)*..... 86
 - 5.16.5.6.1 Request..... 86
 - 5.16.5.6.2 Response..... 87
- 5.16.5.7 *Example 7: unsuccessful charge amount for amount reservation because of denial/refusal by back-end system (Informative)* 87
 - 5.16.5.7.1 Request..... 87
 - 5.16.5.7.2 Response..... 88
- 5.16.6 DELETE 88
- 5.17 RESOURCE: ALL VOLUME RESERVATION TRANSACTIONS FOR AN END USER..... 88**
 - 5.17.1 Request URI variables 88
 - 5.17.2 Response Codes 89
 - 5.17.2.1 *HTTP Response Codes* 89
 - 5.17.2.2 *Exception fault codes*..... 89
 - 5.17.3 GET..... 89
 - 5.17.3.1 *Example: get all volume reservation transactions (Informative)*..... 89
 - 5.17.3.1.1 Request..... 89
 - 5.17.3.1.2 Response..... 89
 - 5.17.4 PUT..... 90
 - 5.17.5 POST..... 90
 - 5.17.5.1 *Example 1: create reserve volume (Informative)*..... 91
 - 5.17.5.1.1 Request..... 91
 - 5.17.5.1.2 Response..... 91
 - 5.17.5.2 *Example 2: create reserve volume with invalid (non-existing) endUserID (Informative)* 92
 - 5.17.5.2.1 Request..... 92
 - 5.17.5.2.2 Response..... 92
 - 5.17.6 DELETE 93
- 5.18 RESOURCE: INDIVIDUAL VOLUME RESERVATION TRANSACTION FOR AN END USER..... 93**
 - 5.18.1 Request URI variables 93
 - 5.18.2 Response Codes 93
 - 5.18.2.1 *HTTP Response Codes* 93
 - 5.18.2.2 *Exception fault codes*..... 93
 - 5.18.3 GET..... 93
 - 5.18.3.1 *Example: get volume reservation (Informative)* 93
 - 5.18.3.1.1 Request..... 93
 - 5.18.3.1.2 Response..... 94
 - 5.18.4 PUT..... 94
 - 5.18.5 POST..... 94
 - 5.18.5.1 *Example: charge volume (Informative)* 94
 - 5.18.5.1.1 Request..... 94
 - 5.18.5.1.2 Response..... 95
 - 5.18.6 DELETE 95
- 5.19 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME RESERVATION TRANSACTION FOR AN END USER 96**
 - 5.19.1 Request URI variables 96
 - 5.19.2 Response Codes 96
 - 5.19.2.1 *HTTP Response Codes* 96
 - 5.19.2.2 *Exception fault codes*..... 96
 - 5.19.3 GET..... 96
 - 5.19.3.1 *Example: get amount for volume reservation (Informative)*..... 96
 - 5.19.3.1.1 Request..... 96
 - 5.19.3.1.2 Response..... 97
 - 5.19.4 PUT..... 97
 - 5.19.5 POST..... 97
 - 5.19.6 DELETE 97
- 5.20 RESOURCE: AMOUNT CONVERTED FROM GIVEN VOLUME 97**
 - 5.20.1 Request URI variables 97
 - 5.20.2 Response Codes 98
 - 5.20.2.1 *HTTP Response Codes* 98

5.20.2.2	Exception fault codes.....	98
5.20.3	GET.....	98
5.20.3.1	Example: get amount converted from volume (Informative).....	98
5.20.3.1.1	Request.....	98
5.20.3.1.2	Response.....	98
5.20.4	PUT.....	99
5.20.5	POST.....	99
5.20.6	DELETE.....	99
6.	FAULT DEFINITIONS.....	100
6.1	6.1 SERVICE EXCEPTIONS.....	100
6.2	6.2 POLICY EXCEPTIONS.....	100
6.2.1	POL0252: Refund request failed.....	100
6.2.2	POL0253: Payment operation refused by user.....	100
APPENDIX A.	CHANGE HISTORY (INFORMATIVE).....	101
A.1	APPROVED VERSION HISTORY.....	101
A.2	DRAFT VERSION 1.0 HISTORY.....	101
APPENDIX B.	STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....	102
B.1	SCR FOR PARLAYREST.PAYMENT SERVER.....	102
B.1.1	SCR for ParlayREST.Payment.Transactions Server.....	102
B.1.2	SCR for ParlayREST.Payment.AmountCharge.Transactions Server.....	102
B.1.3	SCR for ParlayREST.Payment.SplitAmount.Transactions Server.....	103
B.1.4	SCR for ParlayREST.Payment.Individual.AmountCharge Server.....	103
B.1.5	SCR for ParlayREST.Payment.Individual.SplitAmount Server.....	103
B.1.6	SCR for ParlayREST.Payment.VolumeCharge.Transactions Server.....	104
B.1.7	SCR for ParlayREST.Payment.Split.Volume.Transactions Server.....	104
B.1.8	SCR for ParlayREST.Payment.Individual.VolumeCharge Server.....	104
B.1.9	SCR for ParlayREST.Payment.Individual.Split.VolumeCharge Server.....	104
B.1.10	SCR for ParlayREST.Payment.Individual.Amount.VolumeCharge Server.....	105
B.1.11	SCR for ParlayREST.Payment.Individual.Amount.Split.VolumeCharge Server.....	105
B.1.12	SCR for ParlayREST.Payment.Amount.Reserve.Transactions Server.....	105
B.1.13	SCR for ParlayREST.Payment.Amount.Individual.Reserve.Transactions Server.....	106
B.1.14	SCR for ParlayREST.Payment.Volume.Reserve.Transactions Server.....	106
B.1.15	SCR for ParlayREST.Payment.Volume.Individual.Reserve.Transactions Server.....	107
B.1.16	SCR for ParlayREST.Payment.Individual.Amount.Volume Server.....	107
B.1.17	SCR for ParlayREST.Payment.Amount.Converted Server.....	107
APPENDIX C.	APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR SELECTED OPERATIONS (NORMATIVE).....	108
C.1	CHARGE AN AMOUNT.....	108
C.1.1	Example (Informative).....	109
C.1.1.1	Request.....	109
C.1.1.2	Response.....	110
C.2	REFUND AN AMOUNT.....	110
C.2.1	Example (Informative).....	112
C.2.1.1	Request.....	112
C.2.1.2	Response.....	113
C.3	RESERVE AN AMOUNT.....	113
C.3.1	Example (Informative).....	116
C.3.1.1	Request.....	116
C.3.1.2	Response.....	116
C.4	RESERVE AN ADDITIONAL AMOUNT.....	117
C.4.1	Example (Informative).....	119
C.4.1.1	Request.....	119
C.4.1.2	Response.....	120
C.5	CHARGE TO A RESERVATION.....	120
C.5.1	Example (Informative).....	121
C.5.1.1	Request.....	121

C.5.1.2	Response	122
C.6	RELEASE A RESERVATION	122
C.6.1	Example (Informative).....	124
C.6.1.1	Request.....	124
C.6.1.2	Response.....	124
APPENDIX D.	JSON EXAMPLES (INFORMATIVE)	125
D.1	GET ALL TRANSACTIONS (SECTION 5.4.3.1)	125
D.2	REQUEST WITH INVALID ENDUSERID (SECTION 5.4.3.2)	129
D.3	GET ALL AMOUNT TRANSACTIONS (SECTION 5.5.3.1).....	129
D.4	CREATE CHARGE AMOUNT (SECTION 5.5.5.1).....	131
D.5	CREATE REFUND AMOUNT (SECTION 5.5.5.2).....	132
D.6	CLIENT RETRIES POST WITH SAME CLIENTCORRELATOR (SECTION 5.5.5.3).....	133
D.7	UNSUCCESSFUL CHARGE REQUEST BECAUSE OF DENIAL/REFUSAL BY BACK-END SYSTEM (SECTION 5.5.5.4)	133
D.8	GET ALL AMOUNT SPLIT TRANSACTIONS (SECTION 5.6.3.1).....	134
D.9	CREATE SPLIT CHARGE AMOUNT (SECTION 5.6.5.1).....	136
D.10	GET AMOUNT CHARGE (SECTION 5.7.3.1).....	137
D.11	GET AMOUNT SPLIT CHARGE (SECTION 5.8.3.1).....	137
D.12	GET ALL VOLUME CHARGE AND REFUND TRANSACTIONS (SECTION 5.9.3.1).....	138
D.13	CREATE CHARGE VOLUME, RETURNING A REPRESENTATION OF CREATED RESOURCE (SECTION 5.9.5.1).....	139
D.14	CREATE CHARGE VOLUME, RETURNING THE LOCATION OF CREATED RESOURCE (SECTION 5.9.5.2).....	140
D.15	CREATE REFUND VOLUME (SECTION 5.9.5.3).....	141
D.16	GET ALL VOLUME SPLIT CHARGE TRANSACTIONS (SECTION 5.10.3.1)	142
D.17	CREATE VOLUME SPLIT CHARGE (SECTION 5.10.5.1)	143
D.18	GET VOLUME CHARGE (SECTION 5.11.3.1).....	144
D.19	GET VOLUME SPLIT CHARGE (SECTION 5.12.3.1).....	145
D.20	GET AMOUNT FOR VOLUME CHARGE (SECTION 5.13.3.1).....	146
D.21	GET AMOUNT FOR VOLUME SPLIT CHARGE (SECTION 5.14.3.1).....	146
D.22	GET ALL AMOUNT RESERVATION TRANSACTIONS (SECTION 5.15.3.1).....	146
D.23	CREATE RESERVE AMOUNT (SECTION 5.15.5.1).....	148
D.24	GET AMOUNT RESERVATION (SECTION 5.16.3.1)	149
D.25	CHARGE AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.1).....	150
D.26	REPEAT CHARGE REQUEST WITH SAME REFERENCESEQUENCE FOR AN AMOUNT RESERVATION (SECTION 5.16.5.2)	151
D.27	RELEASE AMOUNT RESERVATION (SECTION 5.16.5.3).....	152
D.28	CHARGE PARTIAL AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.4)	153
D.29	RELEASE REMAINING AMOUNT RESERVATION (SECTION 5.16.5.5)	154
D.30	RESERVE ADDITIONAL AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.6).....	154
D.31	UNSUCCESSFUL CHARGE AMOUNT FOR AMOUNT RESERVATION BECAUSE OF DENIAL/REFUSAL BY BACK-END SYSTEM (SECTION 5.16.5.7).....	155
D.32	GET ALL VOLUME RESERVATION TRANSACTIONS (SECTION 5.17.3.1).....	156
D.33	CREATE RESERVE VOLUME (SECTION 5.17.5.1)	158
D.34	CREATE RESERVE VOLUME WITH INVALID ENDUSERID (SECTION 5.17.5.2).....	159
D.35	GET VOLUME RESERVATION (SECTION 5.18.3.1)	159
D.36	CHARGE VOLUME (SECTION 5.18.5.1)	160
D.37	GET AMOUNT FOR VOLUME RESERVATION (SECTION 5.19.3.1).....	161
D.38	GET AMOUNT CONVERTED FROM VOLUME (SECTION 5.20.3.1).....	161
APPENDIX E.	PARLAY X OPERATIONS MAPPING (INFORMATIVE).....	163
APPENDIX F.	MAPPING OF PARLAYREST RESERVATION TRANSACTIONS TO THE CHARGING ENABLER (INFORMATIVE)	164
F.1	CHARGING MECHANISM.....	164
F.2	DETAILED MAPPING OF PARLAYREST TO CHARGING ENABLER.....	164

Figures

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

Figure 2 Amount charge and refund transaction.....	35
Figure 3 Amount split charge transaction	35
Figure 4 Volume charge and refund transaction	36
Figure 5 Volume split charge transaction.....	37
Figure 6 Amount reservation transaction.....	38
Figure 7 Volume reservation transaction	39
Figure 8 Amount converted from volume transaction	40

Tables

Table 1: Parlay X operations mapping	163
--------------------------------------------	-----

1. Scope

This specification defines a RESTful Payment API using an HTTP protocol binding, based on the similar API defined in [3GPP 29.199-6].

2. References

2.1 Normative References

- [3GPP 29.199-6] 3GPP Technical Specification, “Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (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>
- [RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005, URL:<http://www.ietf.org/rfc/rfc4234.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

2.2 Informative References

- [CHRG TS ONLINE] “OMA Online Charging Interface”, Version 1.1, Open Mobile Alliance™. OMA-TS-Charging_Online-V1_1, URL: <http://www.openmobilealliance.org/>
- [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 ParlayREST API specifications”, Open Mobile Alliance™, OMA-WP-Guidelines_for_ParlayREST_API_specifications-20100305-D, URL:<http://www.openmobilealliance.org/>

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

3.3 Abbreviations

ACR	Anonymous Customer Reference
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
ID	Identifier
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
MSISDN	Mobile Subscriber ISDN Number
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
SCR	Static Conformance Requirements
SOAP	Simple Object Access Protocol
SMS	Short Message Service
TS	Technical Specification
UK	United Kingdom
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The ParlayREST Technical Specification for Payment contains the HTTP protocol binding for the Parlay X Payment 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).

4.1 Version 1.0

Version 1.0 of the Payment API specification supports the following operations:

- Charging an amount, split amount, volume or split volume to an end user's account
- Refunding an amount or volume to an end user' account
- Reserving an amount or volume for an end user's account
- Adding an amount or volume to an existing reservation
- Charging to a previously made reservation
- Releasing funds left in a previously made reservation

4.2 Version 1.1

Version 1.1 of the Payment API specification is a maintenance release.

5. Payment API definition

This section is organized to support a comprehensive understanding of the Payment 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).

Finally, Appendix E lists the Parlay X equivalent method for each supported ParlayREST resource and method combination, where applicable.

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 Payment 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.

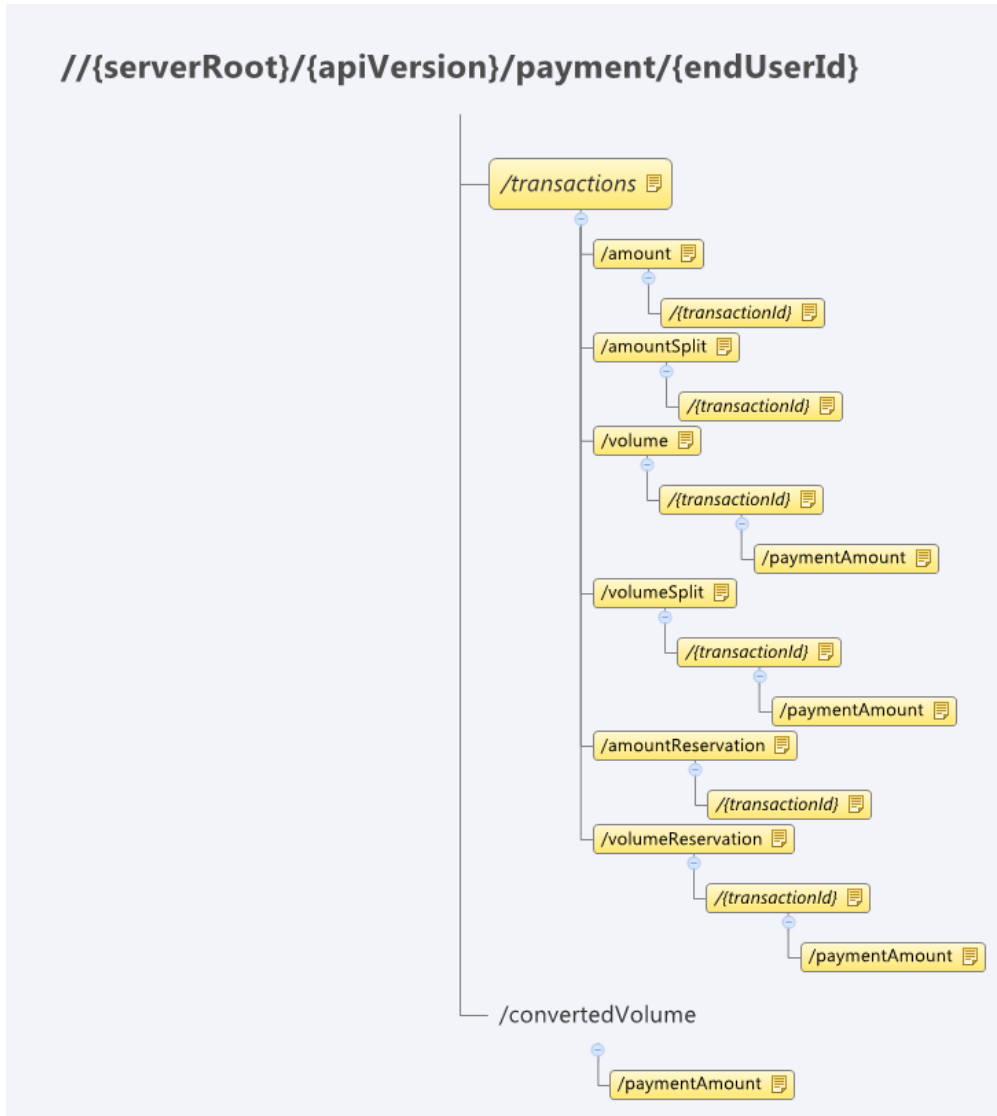


Figure 1 Resource structure defined by this specification

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: All payment transactions

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All end user payment transactions	{endUserId}/transactions	PaymentTransactionList	return all completed and pending payment transactions (amount, volume, amount reservation and volume reservation)	no	no	no

Purpose: Amount charge, split charge, refund transactions

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All amount charge and refund transactions for an end user	{endUserId}/transactions/amount	PaymentTransactionList(used for GET) AmountTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return all amount transactions details for a given end user	no	create new transaction for a given end user	no
All amount split charge transactions for an end user	{endUserId}/transactions/amountSplit	PaymentTransactionList(used for GET) AmountSplitTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return all amount split transactions details for a given end user	no	create new transaction for a given end user	no
Individual amount charge or refund transaction for an end user	{endUserId}/transactions/amount/{transactionId}	AmountTransaction	return amount transaction details (charge, refund)	no	no	no
Individual amount split	{endUserId}/transactions/amountSplit/{transactionId}	AmountSplitTransaction	return amount split	no	no	no

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
charge transaction for an end user	actionId}		transaction details (charge)			

Purpose: Volume charge, split charge, refund transactions

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All volume charge and refund transactions for an end user	{endUserId}/transactions/volume	PaymentTransactionList (used for GET) VolumeTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return all volume transactions details for a given end user	no	create new transaction for a given end user	no
All volume split charge transactions for an end user	{endUserId}/transactions/volumeSplit	PaymentTransactionList(used for GET) VolumeSplitTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return all volume split transactions details for a given end user	no	create new transaction for a given end user	no
Individual volume charge or refund transaction for an end user	{endUserId}/transactions/volume/{transactionId}	VolumeTransaction	return volume transaction details (charge, refund)	no	no	no
Individual volume split charge transaction for an end user	{endUserId}/transactions/volumeSplit/{transactionId}	VolumeSplitTransaction	return volume split transaction details (charge)	no	no	no
Individual amount for volume	{endUserId}/transactions/volume/{transactionId}/paymentAmount	PaymentAmount	return transaction payment information	no	no	no

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
charge or refund transaction for an end user			based on provided volume			
Individual amount for volume split charge transaction for an end user	{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount	PaymentAmount	return transaction payment information for end user share of the volume	no	no	no

Purpose: Amount reservation transactions

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All amount reservation transactions for an end user	{endUserId}/transactions/amountReservation	PaymentTransactionList (used for GET) AmountReservationTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return all amount reservation transactions details for a given end user	no	create new reservation transaction for a given end user	no
Individual amount reservation transaction for an end user	{endUserId}/transactions/amountReservation/{transactionId}	AmountReservationTransaction	return amount reservation transaction details (amount, charge)	no	charge reserved amount, increase reservation amount, release reserved amount	no

Purpose: Volume reservation transactions

Resource	URL Base URL: http://{serverRoot}/{api Version}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE

All volume reservation transactions for an end user	/endUserId/transactions/volumeReservation	PaymentTransactionList (used for GET) VolumeReservationTransaction (used for POST) common:ResourceReference (optional alternative for POST response)	return transaction details for a given end user	no	create new volume reservation transaction for a given end user	no
Individual volume reservation transaction for an end user	/endUserId/transactions/volumeReservation/{transactionId}	VolumeReservationTransaction	return volume reservation transaction details (volume charge)	no	charge reserved volume, increase reserved volume or release reserved volume	no
Individual amount for volume reservation transaction for an end user	/endUserId/transactions/volumeReservation/{transactionId}/paymentAmount	PaymentAmount	return transaction payment information (calculates charge amount based on volume)	no	no	no

Purpose: Amount converted from volume transaction

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/payment	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Amount converted from given volume	/endUserId/convertedVolume/paymentAmount?volume={volume}&unit={Minutes Bytes ...}&contract={contractId}&service={serviceId}&operation={operationId}	PaymentAmount	return the amount resulting from converting the given volume	no	no	no

5.2 Payment ParlayREST API Data Structures

The namespace for the Payment data types is:

urn:oma:xml:rest:payment: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: PaymentTransactionList

List of all payment transactions for query

Element	Type	Optional	Description
amountTransaction	AmountTransaction [0..unbounded]	Yes	Collection of AmountTransaction
amountSplitTransaction	AmountSplitTransaction [0..unbounded]	Yes	Collection of AmountSplitTransaction
volumeTransaction	VolumeTransaction [0..unbounded]	Yes	Collection of VolumeTransaction
volumeSplitTransaction	VolumeSplitTransaction [0..unbounded]	Yes	Collection of VolumeSplitTransaction
volumeReservationTransaction	VolumeReservationTransaction [0..unbounded]	Yes	Collection of VolumeReservationTransaction
amountReservationTransaction	AmountReservationTransaction [0..unbounded]	Yes	Collection of AmountReservationTransaction
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	Provided by the server and points to other resources that are in relationship with the current resource

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

5.2.2 Type: AmountTransaction

Transaction for charge or refund amount

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The end user's account to be charged
paymentAmount	PaymentAmount	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.

serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
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>
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	Provided by the server and points to other resources that are in relationship with the current resource
------	---------------------------	-----	---------------------------------------------------------------------------------------------------------

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

Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.3 Type: AmountSplitTransaction

Transaction for charge or refund split amount

Element	Type	Optional	Description
endUserShare	EndUserShare [1..unbounded]	No	This end user's share of the split charge.
paymentAmount	PaymentAmount	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client

			<p>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>
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	Provided by the server and points to other resources that are in relationship with the current resource

A root element named amountSplitTransaction of type AmountSplitTransaction is allowed in request and/or response bodies. Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.4 Type: VolumeTransaction

Transaction for charge or refund volume

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The end user's account to be charged
paymentVolume	PaymentVolume	No	Information on the volume charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the

			server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
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	Provided by the server and points to other resources that are in relationship with the current resource

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

Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.5 Type: VolumeSplitTransaction

Transaction for charge or refund split volume

Element	Type	Optional	Description
endUserShare	EndUserShare [1..unbounded]	No	This end user's share of the split charge
paymentVolume	PaymentVolume	No	Information on the volume charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a

			<p>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>
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	Provided by the server and points to other resources that are in relationship with the current resource

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

Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.6 Type: AmountReservationTransaction

Transaction for reserve or release amount

Element	Type	Optional	Description
endUserId	xsd:anyURI	Yes	The end user's account to be charged. It MUST be present in the initial reservation request.
paymentAmount	PaymentAmount	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	reserved, charged or released
referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge

			<p>reserved amount (seq=3), etc).</p> <p>The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).</p>
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes.
originalServerReferenceCode	xsd:string	Yes	This may be used to reconcile a current transaction request with an original transaction.
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>

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	Provided by the server and points to other resources that are in relationship with the current resource

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

Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.7 Type: VolumeReservationTransaction

Transaction for reserve or release volume

Element	Type	Optional	Description
endUserId	xsd:anyURI	Yes	The end user's account to be charged. It MUST be present in the initial reservation request.
paymentVolume	PaymentVolume	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	reserved, charged or released
referenceSequence	xsd:integer	No	Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc). The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).

referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes.
originalServerReferenceCode	xsd:string	Yes	This may be used to reconcile a current transaction request with an original transaction.
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>
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	Provided by the server and points to other resources that are in relationship with the current resource

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

Note that the clientCorrelator is used for purposes of error recovery as specified in section 5.6.1 of [REST_TS_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of

the URL of the resource. Section 5.6.1 of [REST_TS_Common] provides a recommendation regarding the generation of the value of this field.

5.2.8 Type: PaymentAmount

Payment amount specific information

Element	Type	Optional	Description
chargingInformation	common:ChargingInformation	No	Holds the charge with amount, currency and description text.
totalAmountCharged	xsd:decimal	Yes	The total amount which has been charged
totalAmountRefunded	xsd:decimal	Yes	The total amount which has been refunded
amountReserved	xsd:decimal	Yes	The amount which has been reserved
chargingMetaData	ChargingMetaData	Yes	Metadata about the charging, such as e.g. information about the merchant, the product, taxation, etc

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

5.2.9 Type: PaymentVolume

Volume amount specific information

Element	Type	Optional	Description
billingText	xsd:string	No	Textual information to appear on the bill
volume	xsd:decimal	No	The volume to be charged
ratingParameter	RatingParameter [0..unbounded]	Yes	Parameters to use when performing rating ("unit", "contract", "service", "operation", etc). There is a maximum of one instance of each parameter type. Example, for the request "reserve 5 minutes of the gold video service", the volume part value is 5 and the parameters part has the following properties: <ul style="list-style-type: none"> • "unit"=minutes • "contract"=gold • "service"=video
totalVolumeCharged	xsd:decimal	Yes	The total volume which has been charged
totalVolumeRefunded	xsd:decimal	Yes	The total volume which has been refunded
volumeReserved	xsd:decimal	Yes	The volume which has been reserved
chargingMetaData	ChargingMetaData	Yes	Metadata about the charging, such as e.g. information about the merchant, the product, taxation, etc

5.2.10 Type: ChargingMetaData

Information about charging data

Element	Type	Optional	Description
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by. This provides visibility to the true merchant in an aggregation scenario and is key in dealing with customer queries where an aggregator submits requests on behalf of another entity.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased. A standard list of category codes would be helpful together with the ability to extend as required. This provides multiple uses including correct taxation, service blocking and service based spending limits.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required. This is useful if the operator needs to interact with the subscriber to authorise the charge (advice of charge) and provides details on how such interaction should be done. Schemes such as PayForIt in the UK show the value in centrally controlled Advice Of Charge and the channel is an important data element.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system. It is important to know if the amount submitted has been pre taxed so the subscribers do not get double taxed on their bill.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies. Allows operators to track charges and group them based on the subscription service which they belong to. Also allows operators to block subscription payments if they have been requested to by the subscriber. For one off purchases, it allows the separate gathering of user consent and provides non repudiation for purchases.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased. This field could contain (for example) the short code of the service or an internal partner service ID and will allow the purchase to be tied directly back to their service catalog. This will be especially important in the US carriers to track payments to specific mobile campaigns.
productId	xsd:string	Yes	Combines with the service ID to uniquely identify the product being purchased. For example if the service ID relates to a music service, the product ID can specify the song. If service ID relates to a short code, the product ID can specify the service on that short code (where multiple services are running on the same short code). This provides an additional layer of visibility to the operator and subscriber and allows for detailed reporting.

5.2.11 Type: EndUserShare

Information for splitting charges between accounts

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The end user's account to be charged
percent	xsd: integer	No	The percentage of this end user's share. The sum of all shares must equal 100. The value of percentage should be positive.

5.2.12 Type: RatingParameter

Rating information for volume payment

Element	Type	Optional	Description
name	xsd:string	No	Free string (e.g.: "unit", "contract", "service", "operation", etc)
value	xsd:string	No	Value of the rating parameter (e.g. "minutes" for "unit", "gold" for "contract", "video" for "service", etc)

5.2.13 Enumeration: TransactionOperationStatus

Enumeration of supported status resulting from transaction operations

Enumeration	Description
Charged	In the request: charge the amount or volume. In the response: The amount or volume has been charged
Refunded	In the request: refund the amount or volume. In the response: The amount or volume has been refunded
Reserved	In the request: reserve the amount or volume. In the response: The amount or volume has been reserved
Released	In the request: release the reservation. In the response: The reservation is released (implicitly by expiration or explicitly by release operation)
Denied	This value only occurs in responses to a GET operation and means that the operation has been denied by the back-end payment server because of an issue with the user's account (e.g. insufficient balance, potential fraud flag on the user's account)
Refused	This value only occurs in responses to a GET operation and means that the operation has been refused (i.e. rejected by end user)

TransactionOperationStatus values in the request body represent the desired successful outcome of the operation. In the response it represents the status of this operation.

TransactionOperationStatus represents the final status of an operation of a transaction on the Server side, as it is stored in the resource representation on the server

An operation can be denied or refused, i.e. the transaction has been unsuccessful. In such a case the Server SHALL return an HTTP 400 Bad Request status and a RequestError data structure, with an indication of a ServiceException SVC0270 (Charge failed) , or either PolicyException POL0252 (Refund request failed) or POL0253 (Payment operation refused by user) as answer to the POST request. The RequestError body MAY include a link to the previously created resource. The resource SHALL be updated to reflect the correct transactionOperationStatus (denied or refused), that can be retrieved using a GET operation on the resource indicated by the link.

5.2.14 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):

- PaymentTransactionList
- AmountTransaction
- AmountSplitTransaction
- AmountReservationTransaction
- VolumeTransaction
- VolumeSplitTransaction
- VolumeReservationTransaction
- PaymentAmount

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

5.3 Sequence diagrams

5.3.1 Amount charge and refund transaction

This figure below shows a scenario to create an amount charge or refund transaction for an end user.

The resource:

- To create an amount charge or refund transaction, create new resource under
`http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount`

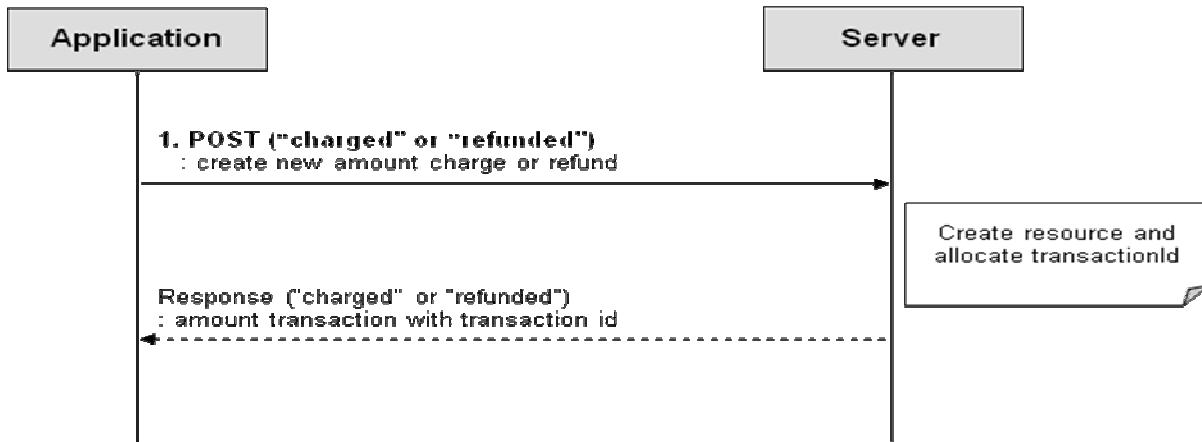


Figure 2 Amount charge and refund transaction

Outline of flow:

1. An application asks for the creation of new amount charge or refund transaction for an end user using **POST** and receives the response with a resource URL containing the transactionId.

5.3.2 Amount split charge transaction

This figure below shows a scenario to create an amount split charge transaction for an end user.

The resource:

- To create an amount split charge transaction for an end user, create new resource under

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit

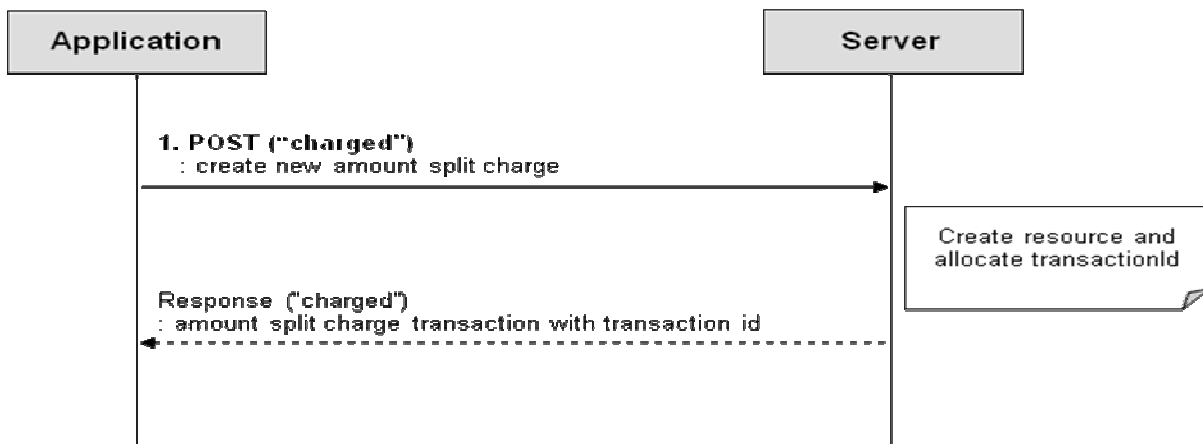


Figure 3 Amount split charge transaction

Outline of flow:

1. An application asks for the creation of new amount split charge transaction for an end user using **POST** and receives the response with a resource URL containing the transactionId.

5.3.3 Volume charge and refund transaction

This figure below shows a scenario to create a volume charge or refund transaction for an end user.

The resource:

- To create a volume charge or refund transaction for an end user, create new resource under **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume**

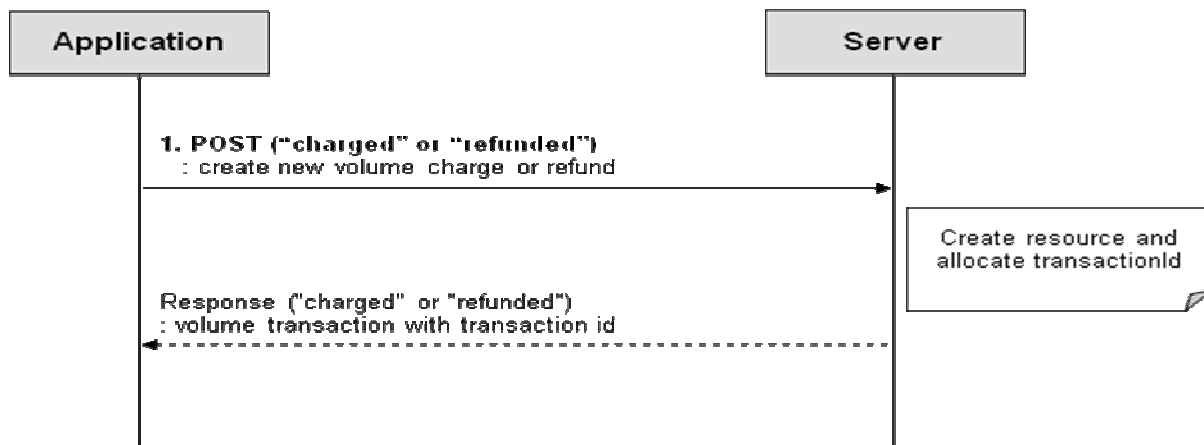


Figure 4 Volume charge and refund transaction

Outline of flow:

1. An application asks for creation of new volume charge or refund transaction for an end user using **POST** and receives the response with a resource URL containing the transactionId.

5.3.4 Volume split charge transaction

This figure below shows a scenario to create a volume split charge transaction for an end user.

The resource and operation used

- To create a volume split charge transaction for an end user, create new resource under **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit**

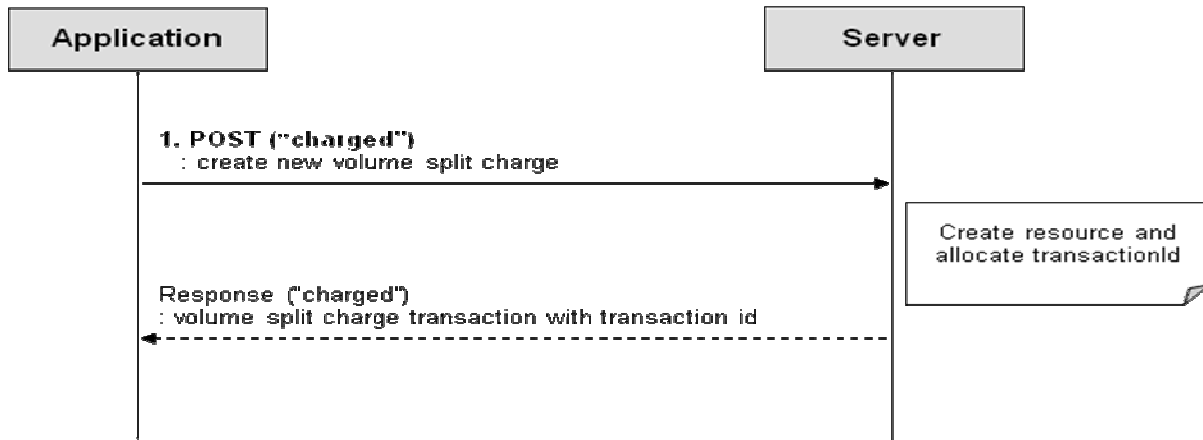


Figure 5 Volume split charge transaction

Outline of flow:

1. An application asks for creation of new volume split charge transaction for an end user using **POST** and receives the response with a resource URL containing the transactionId.

5.3.5 Amount reservation transaction

This figure below shows a scenario to create, to charge and to release an amount reservation transaction for an end user.

The resources:

- To create an amount reservation transaction for an end user, create new resource under
`http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation`
- To update an amount reservation transaction for an end user, use resource
`http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}`

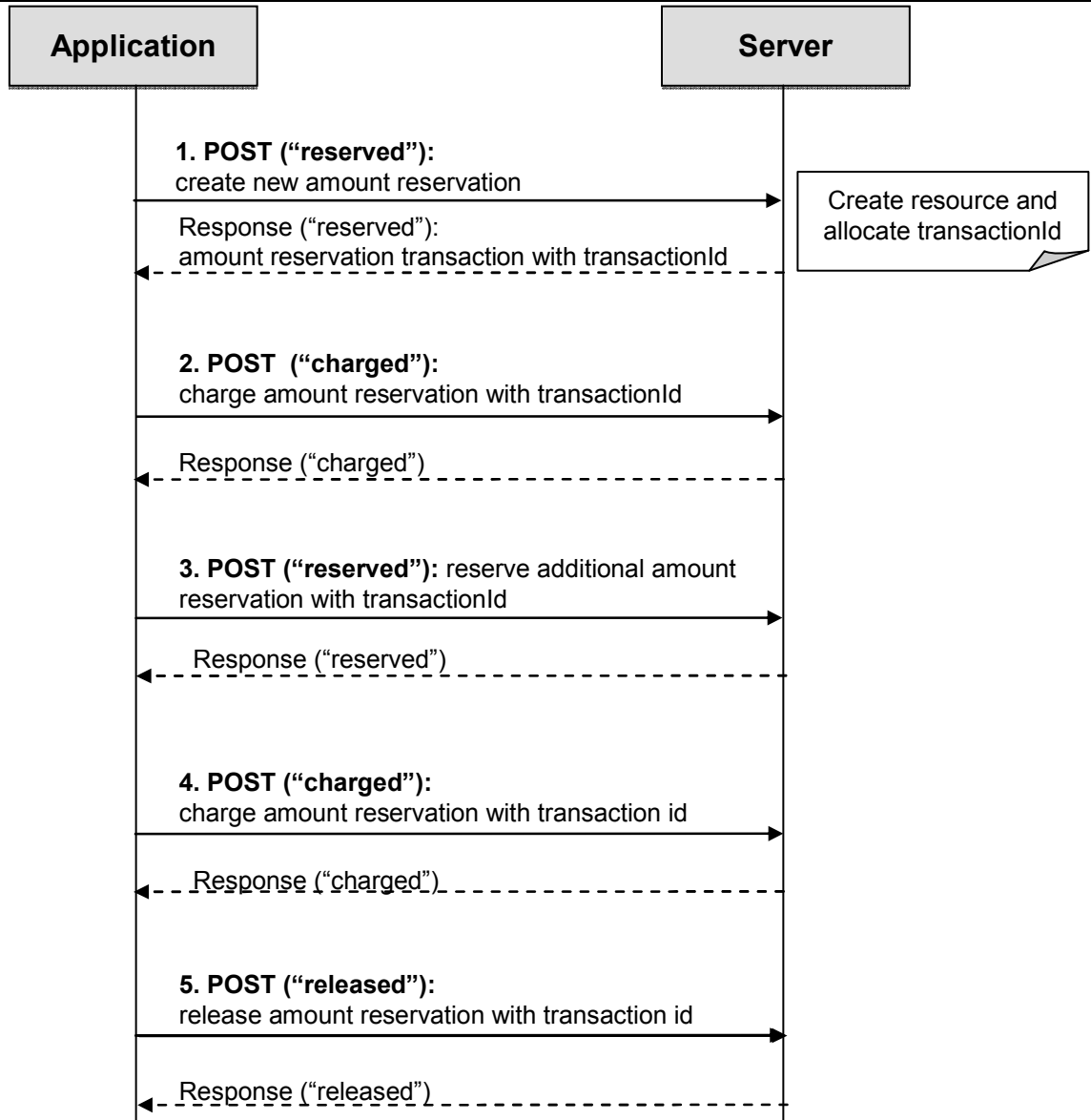


Figure 6 Amount reservation transaction

Outline of flow:

1. Application creates a new amount reservation transaction for an end user. **(POST)**
2. Application charges an amount reservation with transactionId. **(POST)**
3. In case that add to the existing amount reservation, application reserves an additional amount reservation with transactionId. **(POST)**
4. Application charges an amount reservation with transactionId. **(POST)**
5. Application releases an amount reservation transaction for an end user. **(POST)**

5.3.6 Volume reservation transaction

This figure below shows a scenario to create, to charge and to release a volume reservation transaction for an end user.

The resource:

- To create a volume reservation transaction for an end user, create new resource under **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation**
- To update a volume reservation transaction for an end user, use resource **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}**

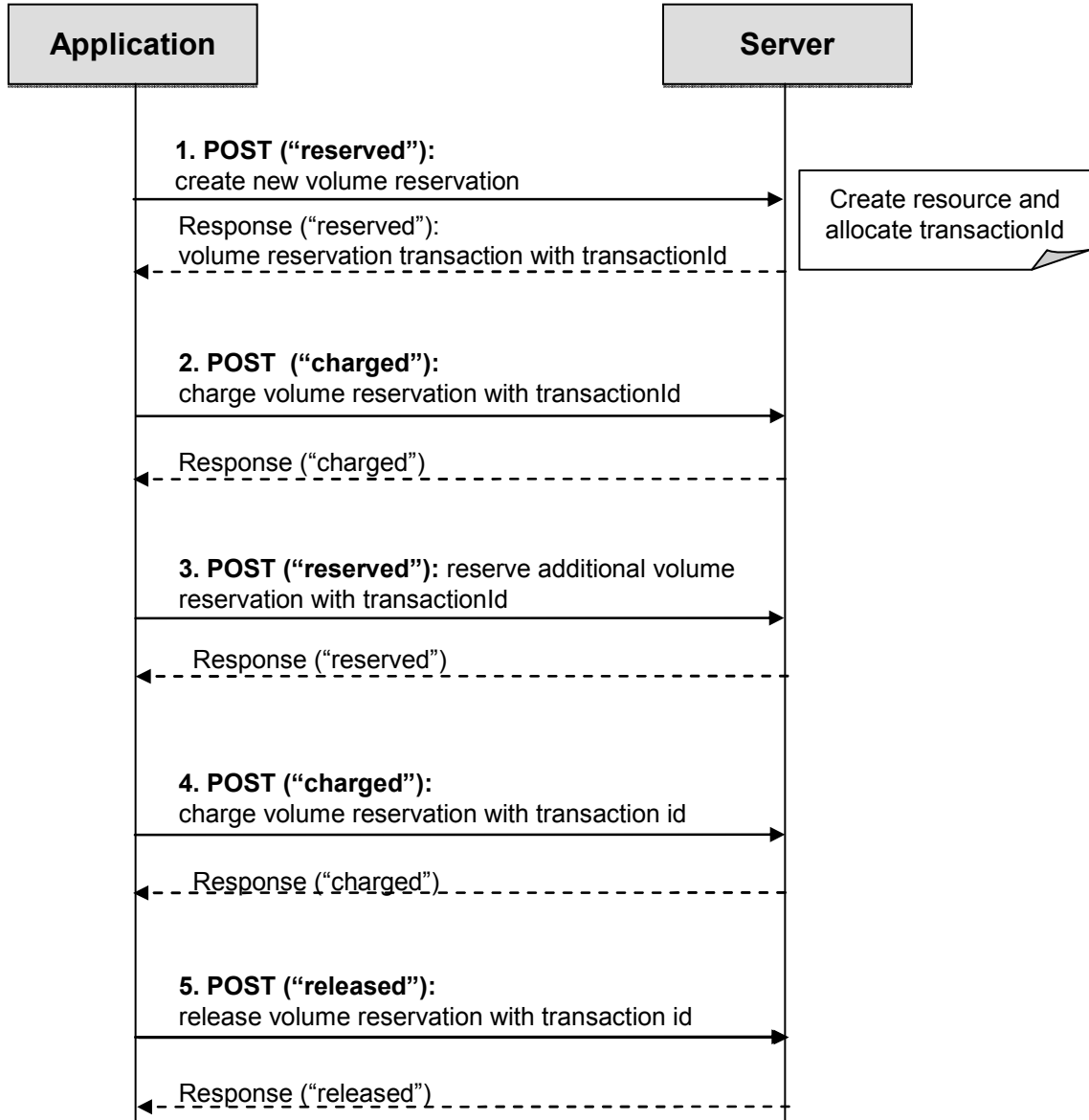


Figure 7 Volume reservation transaction

Outline of flow:

1. Application creates a new volume reservation transaction for an end user. (POST)
2. Application charges a volume reservation with transactionId. (POST)

3. In case that add to the existing volume reservation, application reserves an additional volume reservation with transactionId. (POST)
4. Application charges a volume reservation with transactionId. (POST)
5. Application releases a volume reservation transaction for an end user. (POST)

5.3.7 Amount converted from volume transaction

This figure below shows a scenario to provide access to amount converted from given volume.

The resource (illustrated with optional URL parameters) and operation used

- To get the amount for a given volume use the resource below with mandatory URL parameter “ volume” and applicable optional URL parameters (e.g. “ unit” , “ contract” , “ service” , “ operation”)

http://{serverRoot}/{apiVersion}/payment/{endUserId}/convertedVolume/paymentAmount

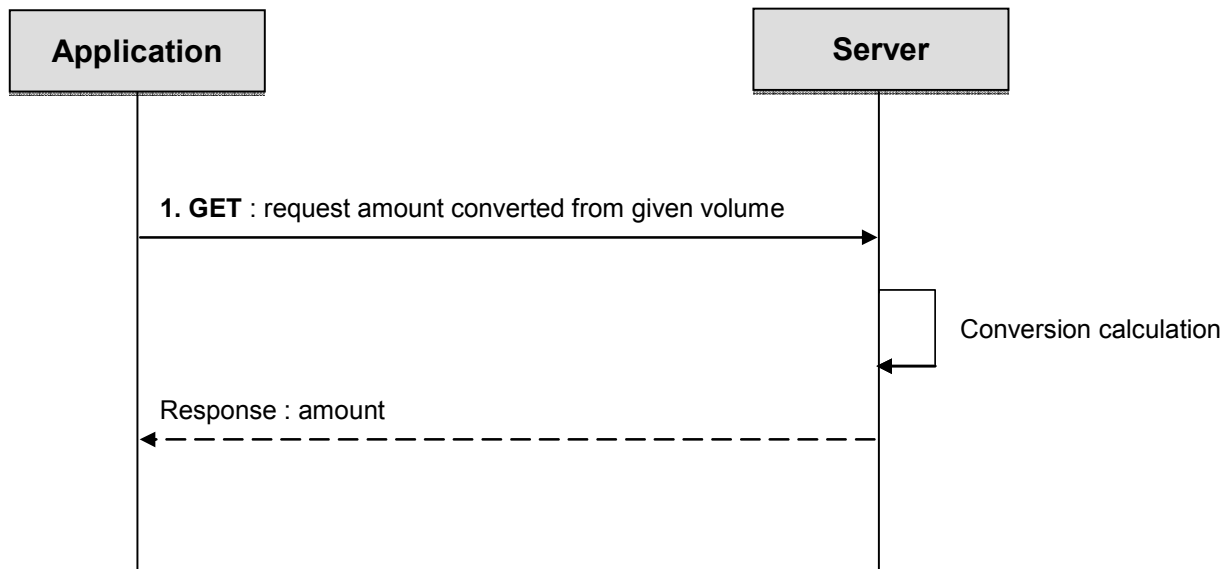


Figure 8 Amount converted from volume transaction

Outline of flow:

1. An application requests the amount resulting from converting the given volume for an end user and receives the amount information.

5.4 Resource: All payment transactions for an end user

The resource used is: **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions**

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

This resource is used to provide access all completed and pending transactions (amount, volume, amount reservation and volume reservation) for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.4.2 Response Codes

5.4.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.4.3 GET

This operation is used to obtain all completed and pending payment transactions (amount, volume, amount reservation and volume reservation) for an end user.

5.4.3.1 Example 1: get all transactions (Informative)

5.4.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions 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: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <!-- COMPLETED AMOUNT CHARGE TRANSACTION -->
  <amountTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
    <chargingInformation>
    <description>Test amount transaction "Charged"</description>
```

```

    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
</totalAmountCharged>10</totalAmountCharged>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT REFUND TRANSACTION -->
<amountTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-456</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54322</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans456</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
<amountSplitTransaction>
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0101</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-789</serverReferenceCode>
  <clientCorrelator>54323</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans789</resourceURL>

```

```

</amountSplitTransaction>
<!-- COMPLETED VOLUME CHARGE TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>DEF-123</serverReferenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124</resourceURL>
</volumeTransaction>
<!-- COMPLETED VOLUME REFUND TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Refunded"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>GHI-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54324</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans125</resourceURL>
</volumeTransaction>
<!-- COMPLETED VOLUME SPLIT CHARGE TRANSACTION -->
<volumeSplitTransaction>
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0101</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>

```

```

</paymentVolume>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>JKL-123</serverReferenceCode>
<clientCorrelator>54325</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeSplit/trans126</resourceURL>
</volumeSplitTransaction>
<!--
  VOLUME RESERVATION TRANSACTION "RESERVED"
-->
<volumeReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans999</resourceURL>
</volumeReservationTransaction>
<!-- VOLUME RESERVATION TRANSACTION "CHARGED"-->
<volumeReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>MNO-123</serverReferenceCode>
  <clientCorrelator>66667</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans127</resourceURL>

</volumeReservationTransaction>
<!--
  COMPLETED AMOUNT RESERVATION TRANSACTION
-->

```

```

<amountReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged"</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>25</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>PQR-123</serverReferenceCode>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans128</resourceURL>

</amountReservationTransaction>
<!-- AMOUNT RESERVATION TRANSACTION "RESERVED"-->
<amountReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <!-- last additional reserve applied -->
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans998</resourceURL>
</amountReservationTransaction>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions</resourceURL>
</payment:paymentTransactionList>

```

5.4.3.2 Example 2: request with invalid (non-existing) endUserId (Informative)

5.4.3.2.1 Request

```

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

```

5.4.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="PaymentTransactionList"
    href="http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions"/>
  <serviceException>
    <messageId>SVC0004</messageId>
    <text>Invalid input value. The address %1 does not exist.</text>
    <variables>tel:+1-555-555-0100</variables>
  </serviceException>
</common:requestError>
```

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: All amount charge and refund transactions for an end user

The resource used is: **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount**

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

This resource is used to provide access to all the amount charge and refund transactions for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.5.2 Response Codes

5.5.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.5.3 GET

This operation is used to obtain all amount charge and refund transactions for an end user.

5.5.3.1 Example: get all amount transactions (Informative)

5.5.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount 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: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <!-- COMPLETED AMOUNT CHARGE TRANSACTION -->
  <amountTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction "Charged"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
  </serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
</amountTransaction>
  <!-- COMPLETED AMOUNT CHARGE TRANSACTION - ACRed end user id-->
  <amountTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
      <chargingInformation>
```

```

<description>Test amount transaction "Charged"</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountCharged>10</totalAmountCharged>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>DEF-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans124</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT REFUND TRANSACTION -->
<amountTransaction>
<endUserId>tel:+1-555-555-0100</endUserId>
<paymentAmount>
<chargingInformation>
<description>Test amount transaction "Refunded"</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountRefunded>10</totalAmountRefunded>
</paymentAmount>
<transactionOperationStatus>Refunded</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>WXY-123</serverReferenceCode>
<originalServerReferenceCode>DEF-123</originalServerReferenceCode>
<clientCorrelator>54330</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans129</resourceURL>
</amountTransaction>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount</resourceURL>
</payment:paymentTransactionList>

```

5.5.4 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].

5.5.5 POST

This operation is used to create a new transaction for an end user.

5.5.5.1 Example 1: create charge amount

(Informative)

5.5.5.1.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml

```


Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
```

5.5.5.1.2 Response

HTTP/1.1 201 Created
 Content-Type: application/xml
 Content-Length: nnnn
 Date: Thu, 04 Jun 2009 02:51:59 GMT
 Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

5.5.5.2 Example 2: create refund amount

(Informative)

5.5.5.2.1 Request

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

```

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

```

5.5.5.2.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans130

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>XYZ-123</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans130</resourceURL>
</payment:amountTransaction>

```

5.5.5.3 Example 3: client retries POST with same clientCorrelator (Informative)

Note: This example illustrates the case where the resource was already successfully created by the previous POST request, but the "201 Created" response got lost. See [OMA_REST_TS_Common] section 5.6.1 for other possible cases.

5.5.5.3.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
```

5.5.5.3.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
</resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
```

```
</payment:amountTransaction>
```

5.5.5.4 Example 4: unsuccessful charge request because of denial/refusal by back-end system (Informative)

This example illustrates the case where a resource will not be created, because of denial or refusal by back-end system.

5.5.5.4.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
```

5.5.5.4.2 Response

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <serviceException>
    <messageId>SVC0270</messageId>
    <text>Charging operation failed, the charge was not applied.</text>
  </serviceException>
</common:requestError>
```

5.5.6 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].

5.6 Resource: All amount split charge transactions for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit

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

This resource is used to provide access to all the amount split charge transactions for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.6.2 Response Codes

5.6.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.6.3 GET

This operation is used to obtain all amount split charge transactions for an end user.

5.6.3.1 Example: get all amount split transactions (Informative)

5.6.3.1.1 Request

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

5.6.3.1.2 Response

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

Date: Thu, 04 Jun 2009 02:51:59 GMT

```

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
  <amountSplitTransaction>
    <endUserShare>
      <endUserId>tel:+1-555-555-0100</endUserId>
      <percent>30</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+1-555-555-0101</endUserId>
      <percent>70</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction "Charged"</description>
        <currency>USD</currency>
        <amount>20</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>20</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
  </serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55552</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123</resourceURL>
</amountSplitTransaction>
  <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
  <amountSplitTransaction>
    <endUserShare>
      <endUserId>tel:+1-555-555-0100</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+1-555-555-0101</endUserId>
      <percent>80</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction "Charged" </description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012346</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12346</referenceCode>
  </serverReferenceCode>DEF-123</serverReferenceCode>
  <clientCorrelator>55553</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans124</resourceURL>
</amountSplitTransaction>

```

```
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit</resourceURL>
</payment:paymentTransactionList>
```

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

5.6.5 POST

This operation is used to create a new transaction for an end user.

5.6.5.1 Example: create split charge amount (Informative)

5.6.5.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT SPLIT CHARGE TRANSACTION -->
<payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>30</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0101</endUserId>
    <percent>70</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54431</clientCorrelator>
</payment:amountSplitTransaction>
```

5.6.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

Location: <http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123>

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT SPLIT CHARGE TRANSACTION -->
<!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
<payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>30</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0101</endUserId>
    <percent>70</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54431</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123</resourceURL>
</payment:amountSplitTransaction>
```

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, POST’ field in the response as per section 14.7 of [RFC 2616].

5.7 Resource: Individual amount charge or refund transaction for an end user

The resource used is:

<http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}>

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

This resource is used to provide access to an individual amount charge or refund transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.7.2 Response Codes

5.7.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.7.3 GET

This operation is used to return individual completed or pending amount charge and refund transaction information for an end user.

5.7.3.1 Example: get amount charge (Informative)

This example shows also an alternative way to indicate desired content type in response from the server, by using URL query parameter “?resFormat” which is described in [OMA_REST_TS_Common]. Note: this example also illustrates how to indicate in the request the expected response body format.

5.7.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123?resFormat=XML HTTP/1.1
Host: example.com
```

5.7.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
  <chargingInformation>
```

```

<description>Test amount transaction "Charged" </description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST-012345</code>
</chargingInformation>
<totalAmountCharged>10</totalAmountCharged>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>

```

5.7.4 PUT

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

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

5.7.6 DELETE

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

5.8 Resource: Individual amount split charge transaction for an end user

The resource used is: `http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}`

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

This resource is used to provide access to an individual amount split charge transaction for an end user.

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: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.8.2 Response Codes

5.8.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.8.3 GET

This operation is used to return individual completed or pending amount split charge transaction information for an end user.

5.8.3.1 Example: get amount split charge (Informative)

5.8.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123
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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- AMOUNT SPLIT CHARGE TRANSACTION -->
  <payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserShare>
      <endUserId>tel:+1-555-555-0100</endUserId>
      <percent>30</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+1-555-555-0101</endUserId>
      <percent>70</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction "Charged" </description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
  </serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
```

```
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123</resourceURL>

</payment:amountSplitTransaction>
```

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: All volume charge and refund transactions for an end user

The resource used is: **http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume**

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

This resource is used to provide access to all the volume charge and refund transactions for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.9.2 Response Codes

5.9.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.9.3 GET

This operation is used to obtain all volume charge and refund transactions for an end user.

5.9.3.1 Example: get all volume charge and refund transactions (Informative)

5.9.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume 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: Thu, 04 Jun 2009 02:51:59 GMT

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

  <!-- COMPLETED VOLUME CHARGE TRANSACTION -->
  <volumeTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction "Charged" </billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>55551</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123</resourceURL>
  </volumeTransaction>
  <!-- COMPLETED VOLUME REFUND TRANSACTION -->
  <volumeTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction "Refunded"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeRefunded>10</totalVolumeRefunded>
    </paymentVolume>
    <transactionOperationStatus>Refunded</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>DEF-123</serverReferenceCode>
```

```

<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
<clientCorrelator>55552</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124</resourceURL>
</volumeTransaction>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume</resourceURL>
</payment:paymentTransactionList>

```

5.9.4 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].

5.9.5 POST

This operation is used to create a new volume transaction for an end user.

5.9.5.1 Example 1: create charge volume, returning a representation of created resource (Informative)

5.9.5.1.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55551</clientCorrelator>
</payment:volumeTransaction>

```

5.9.5.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml

```

```

Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->

```

```

<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123</resourceURL>
</payment:volumeTransaction>

```

5.9.5.2 Example 2: create charge volume, returning the location of created resource (Informative)

5.9.5.2.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55551</clientCorrelator>
</payment:volumeTransaction>

```

5.9.5.2.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```
<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:common:1">
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123</resourceURL>
</common:resourceReference>
```

5.9.5.3 Example 3: create refund volume

(Informative)

5.9.5.3.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME REFUND TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Refunded"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>55552</clientCorrelator>
</payment:volumeTransaction>
```

5.9.5.3.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME REFUND TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentVolume>
    <billingText>Test volume transaction "Refunded"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
```



```

</ratingParameter>
<totalVolumeRefunded>10</totalVolumeRefunded>
</paymentVolume>
<transactionOperationStatus>Refunded</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>DEF-123</serverReferenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
<clientCorrelator>55552</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124</resourceURL>
</payment:volumeTransaction>

```

5.9.6 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].

5.10 Resource: All volume split charge transactions for an end user

The resource used is: `http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit`

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

This resource is used to provide access to all the volume split charge transactions for an end 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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.10.2 Response Codes

5.10.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.10.3 GET

This operation is used to obtain all volume split charge transactions for an end user.

5.10.3.1 Example: get all volume split charge transactions (Informative)

5.10.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit
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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <volumeSplitTransaction>
    <endUserShare>
      <endUserId>tel:+1-555-555-0100</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+1-555-555-0101</endUserId>
      <percent>80</percent>
    </endUserShare>
    <paymentVolume>
      <billingText>Test volume transaction "Charged"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
  </volumeSplitTransaction>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55553</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123</resourceURL>
  <!-- COMPLETED VOLUME SPLIT CHARGE TRANSACTION -->
</payment:paymentTransactionList>
```

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

5.10.5 POST

This operation is used to create a new volume split transaction for an end user.

5.10.5.1 Example: create volume split charge

(Informative)

5.10.5.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME SPLIT CHARGE TRANSACTION -->
<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55553</clientCorrelator>
</payment:volumeSplitTransaction>
```

5.10.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME SPLIT CHARGE TRANSACTION -->
```

```

<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserShare>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+1-555-555-0101</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55553</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123</resourceURL>
</payment:volumeSplitTransaction>

```

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, POST’ field in the response as per section 14.7 of [RFC 2616].

5.11 Resource: Individual volume charge or refund transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}

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

This resource is used to provide access to an individual volume charge or refund transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.11.2 Response Codes

5.11.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.11.3 GET

This operation is used to return individual completed or pending volume charge and refund transaction information for an end user.

5.11.3.1 Example: get volume charge

(Informative)

5.11.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME CHARGE TRANSACTION -->
  <payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction "Charged" </billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123</resourceURL>
  </payment:volumeTransaction>
```

5.11.4 PUT

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

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

5.11.6 DELETE

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

5.12 Resource: Individual volume split charge transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}

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

This resource is used to provide access to an individual volume split charge transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.12.2 Response Codes

5.12.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.12.3 GET

This operation is used to return individual completed or pending volume split charge transaction information for an end user.

5.12.3.1 Example: get volume split charge (Informative)

5.12.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME SPLIT CHARGE TRANSACTION -->
  <payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserShare>
      <endUserId>tel:+1-555-555-0100</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+1-555-555-0101</endUserId>
      <percent>80</percent>
    </endUserShare>

    <paymentVolume>
      <billingText>Test volume transaction "Charged" </billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
  </serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55556</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123</resourceURL>
</payment:volumeSplitTransaction>
```

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 amount for volume charge or refund transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}/paymentAmount

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

This resource is used to provide access to an individual payment amount information based on provided volume transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.13.2 Response Codes

5.13.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.13.3 GET

This operation is used to return individual transaction payment information based on provided volume transaction for an end user.

5.13.3.1 Example: get amount for volume charge

(Informative)

5.13.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/volume/trans999/paymentAmount 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:payment:1">
<chargingInformation>
  <description>10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+1-555-555-0100</description>
  <currency>USD</currency>
  <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

5.13.4 PUT

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

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’ field in the response as per section 14.7 of [RFC 2616].

5.13.6 DELETE

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

5.14 Resource: Individual amount for volume split charge transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount

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

This resource is used to provide access to an individual payment amount information based on provided volume split transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.14.2 Response Codes

5.14.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.14.3 GET

This operation is used to return individual transaction payment information based on provided volume split transaction for an end user.

5.14.3.1 Example: get amount for volume split charge (Informative)

5.14.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans999/paymentAmount 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:payment:1">
<chargingInformation>
  <description>10 Minutes converted to USD for transaction=trans999</description>
  <currency>USD</currency>
  <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

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: All amount reservation transactions for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation

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

This resource is used to provide access to all the completed and pending amount reservation transactions for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.15.2 Response Codes

5.15.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.15.3 GET

This operation is used to return all amount reservation transactions details and state for a given end user.

5.15.3.1 Example: get all amount reservation transactions (Informative)

5.15.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <!-- COMPLETED AMOUNT RESERVATION TRANSACTION -->
  <amountReservationTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>

    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction "Charged" </description>
        <currency>USD</currency>
        <amount>15</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>25</totalAmountCharged>
      <amountReserved>0</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceSequence>2</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans123</resourceURL>
</amountReservationTransaction>
  <!-- AMOUNT RESERVATION TRANSACTION "RESERVED" -->
  <amountReservationTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction "Reserved"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>0</totalAmountCharged>
      <amountReserved>10</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <clientCorrelator>55556</clientCorrelator>
```

```

<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999</resourceURL>
</amountReservationTransaction>
<!--
  AMOUNT RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL RESERVATION AMOUNT APPLIED
-->
<amountReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans998</resourceURL>
</amountReservationTransaction>
<!--
  AMOUNT RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL
  RESERVATION AMOUNT IS APPLIED SECOND TIME
-->
<amountReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>25</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans997</resourceURL>
</amountReservationTransaction>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation</resourceURL>
</payment:paymentTransactionList>

```

5.15.4 PUT

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

5.15.5 POST

This operation is used to create a new transaction for an end user.

5.15.5.1 Example: create amount reservation (Informative)

5.15.5.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com
```

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

```
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>55555</clientCorrelator>
</payment:amountReservationTransaction>
```

5.15.5.1.2 Response

```
HTTP/1.1 201 Created
```

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

```
Content-Length: nnnn
```

```
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
```

```

<code>TEST-012345</code>
</chargingInformation>
<totalAmountCharged>0</totalAmountCharged>
<amountReserved>10</amountReserved>
</paymentAmount>
<transactionOperationStatus>Reserved</transactionOperationStatus>
<referenceSequence>1</referenceSequence>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

5.15.6 DELETE

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

5.16 Resource: Individual amount reservation transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}

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

This resource is used to provide access to an individual amount reservation transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.16.2 Response Codes

5.16.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.16.3 GET

This operation is used to return individual completed or pending amount reservation transaction information for an end user.

5.16.3.1 Example: get amount reservation (Informative)

5.16.3.1.1 Request

```
GET /exampleAPI/1/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999
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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

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

5.16.5 POST

This operation is used to change transaction operation state to “charged” or “released” or “reserved” for an amount reservation transaction for an end user.

5.16.5.1 Example 1: charge amount for amount reservation (informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a charge against this reservation is requested via referenceSequence 2.

5.16.5.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>

  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:amountReservationTransaction>
```

5.16.5.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
```

```

<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

5.16.5.2 Example 2: repeat a charge request with same referenceSequence for an amount reservation transaction (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a charge amount for amount reservation was made via referenceSequence 2, but while the 2nd step succeeded on the Server side, the client did not receive the response from the server due to communication failure. The client is repeating the request, but the server will recognize the same referenceSequence as indication of a repeated request, and will not change the resource, instead just returning the same response as in the previous attempt.

5.16.5.2.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:amountReservationTransaction>

```

5.16.5.2.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>

```

```

<currency>USD</currency>
<amount>10</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountCharged>10</totalAmountCharged>
<amountReserved>0</amountReserved>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

5.16.5.3 Example 3: release amount reservation (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and it is released via referenceSequence 2.

5.16.5.3.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Released"</description>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Released</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
</payment:amountReservationTransaction>

```

5.16.5.3.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">

```

```

<endUserId>tel:+1-555-555-0100</endUserId>
<paymentAmount>
  <chargingInformation>
    <description>Test amount reservation transaction "Released"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>0</amountReserved>
</paymentAmount>
<transactionOperationStatus>Released</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<clientCorrelator>55556</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

5.16.5.4 Example 4: charge partial amount for amount reservation (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a partial charge against this reservation additional is requested via referenceSequence 2.

5.16.5.4.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:amountReservationTransaction>

```

5.16.5.4.2 Response

```

HTTP/1.1 200 OK

```

```

Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>5</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

5.16.5.5 Example 5: release remaining amount reservation (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a charge against this reservation via referenceSequence 2 (POST update), and now a release of the remaining reservation is requested via referenceSequence 3.

5.16.5.5.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Released"</description>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Released</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
</payment:amountReservationTransaction>

```

5.16.5.5.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- AMOUNT RESERVATION TRANSACTION -->
  <payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction "Released"</description>
        <currency>USD</currency>
        <amount>5</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>5</totalAmountCharged>
      <amountReserved>0</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Released</transactionOperationStatus>
    <referenceSequence>3</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55558</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
  </payment:amountReservationTransaction>

```

5.16.5.6 Example 6: reserve additional amount for amount reservation (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and an additional reservation is requested via referenceSequence 2.

5.16.5.6.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>

```

```
</payment:amountReservationTransaction>
```

5.16.5.6.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- AMOUNT RESERVATION TRANSACTION -->
  <payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction "Reserved"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>0</totalAmountCharged>
      <amountReserved>15</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>2</referenceSequence>
    <clientCorrelator>55559</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
  </payment:amountReservationTransaction>
```

5.16.5.7 Example 7: unsuccessful charge amount for amount reservation because of denial/refusal by back-end system (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a charge transaction is requested via referenceSequence 2, but is denied because of insufficient funds.

5.16.5.7.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
```

```

<code>TEST012345</code>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
</payment:amountReservationTransaction>

```

5.16.5.7.2 Response

```

HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="AmountReservationTransaction"
    href="http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999"/>
  <serviceException>
    <messageId>SVC0270</messageId>
    <text>Charging operation failed, the charge was not applied.</text>
  </serviceException>
</common:requestError>

```

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, POST’ field in the response as per section 14.7 of [RFC 2616].

5.17 Resource: All volume reservation transactions for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation

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

This resource is used to provide access to all the volume reservation transactions for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.17.2 Response Codes

5.17.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.17.3 GET

This operation is used to obtain all volume reservation transactions for an end user.

5.17.3.1 Example: get all volume reservation transactions (Informative)

5.17.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation 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: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:payment:1">
  <!-- VOLUME RESERVATION TRANSACTION "RESERVED" -->
  <volumeReservationTransaction>
    <endUserId>tel:+1-555-555-0100</endUserId>

    <paymentVolume>
      <billingText>Test volume reservation transaction "Reserved"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>0</totalVolumeCharged>
      <volumeReserved>10</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <clientCorrelator>66666</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999</resourceURL>

  </volumeReservationTransaction>
  <!--
  VOLUME RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL
  VOLUME IS APPLIED
  -->
```

```

<volumeReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>15</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66667</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans998</resourceURL>

</volumeReservationTransaction>
<!-- VOLUME RESERVATION TRANSACTION "CHARGED" -->
<volumeReservationTransaction>
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-22345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>66668</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans123</resourceURL>

</volumeReservationTransaction>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation</resourceURL>
</payment:paymentTransactionList>

```

5.17.4 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].

5.17.5 POST

This operation is used to create a new volume transaction for an end user.

5.17.5.1 Example 1: create reserve volume**(Informative)****5.17.5.1.1 Request**

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME RESERVATION TRANSACTION -->
  <payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume reservation transaction "Reserved"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <clientCorrelator>66666</clientCorrelator>
  </payment:volumeReservationTransaction>

```

5.17.5.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME RESERVATION TRANSACTION -->
  <payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>

    <paymentVolume>
      <billingText>Test volume reservation transaction "Reserved"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>0</totalVolumeCharged>
      <volumeReserved>10</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999</resourceURL>

```

```
</payment:volumeReservationTransaction>
```

5.17.5.2 Example 2: create reserve volume with invalid (non-existing) endUserID (Informative)

5.17.5.2.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME RESERVATION TRANSACTION -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>66667</clientCorrelator>
</payment:volumeReservationTransaction>
```

5.17.5.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <serviceException>
    <messageId>SVC0004</messageId>
    <text>Invalid input value. The address %1 does not exist.</text>
    <variables>tel:+1-555-555-0100</variables>
  </serviceException>
</common:requestError>
```

5.17.6 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].

5.18 Resource: Individual volume reservation transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}

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

This resource is used to provide access to an individual volume reservation transaction for an end user.

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: http://example.com/exampleAPI
apiVersion	version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.18.2 Response Codes

5.18.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.18.3 GET

This operation is used to return individual completed or pending volume reservation transaction information for an end user.

5.18.3.1 Example: get volume reservation (Informative)

5.18.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME RESERVATION TRANSACTION -->
  <payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume reservation transaction "Reserved"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>0</totalVolumeCharged>
      <volumeReserved>10</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>66666</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999</resourceURL>
  </payment:volumeReservationTransaction>

```

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 to change transaction operation state to “charged” or “released” or “reserved” for a volume reservation transaction for an end user.

5.18.5.1 Example: charge volume (Informative)

This example assumes an initial reservation was made via referenceSequence 1 (POST request) and a charge against this reservation is requested via referenceSequence 2.

5.18.5.1.1 Request

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME RESERVATION TRANSACTION -->

```

```

<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Charged" </billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:volumeReservationTransaction>

```

5.18.5.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
  <!-- VOLUME RESERVATION TRANSACTION -->
  <payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
    <endUserId>tel:+1-555-555-0100</endUserId>
    <paymentVolume>
      <billingText>Test volume reservation transaction "Charged" </billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
      <volumeReserved>0</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceSequence>2</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>66666</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans999</resourceURL>
  </payment:volumeReservationTransaction>

```

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 amount for volume reservation transaction for an end user

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount

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

This resource is used to provide access to an individual payment amount information based on provided volume reservation transaction for an end user.

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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)
transactionId	unique transaction identifier

5.19.2 Response Codes

5.19.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.19.3 GET

This operation is used to return individual transaction payment information based on provided volume reservation transaction (calculates charge amount based on volume) for an end user.

5.19.3.1 Example: get amount for volume reservation (Informative)

5.19.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999/paymentAmount 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: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:payment:1">
<chargingInformation>
  <description>10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+1-555-555-0100</description>
  <currency>USD</currency>
  <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>

```

5.19.4 PUT

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

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

5.19.6 DELETE

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

5.20 Resource: Amount converted from given volume

The resource used is:

http://{serverRoot}/{apiVersion}/payment/{endUserId}/convertedVolume/paymentAmount

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

This resource is used to provide access to amount converted from given volume.

5.20.1 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 client wants to use (e.g. 1 for version 1.x)
endUserId	the end user's account (xsd:anyURI)

5.20.2 Response Codes

5.20.2.1 HTTP 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 Payment, see [3GPP 29.199-6] and section 6 of the present document.

5.20.3 GET

This operation is used to return the amount resulting from converting the given volume for an end user.

Request URL parameters are:

Name	Type/Values	Optional	Description
volume	xsd:decimal	No	The given volume to be converted
unit	xsd:string	Yes	Parameter to use when performing rating (e.g. "minutes")
contract	xsd:string	Yes	Parameter to use when performing rating (e.g. number of a contract that may govern the use)
service	xsd: string	Yes	Parameter to use when performing rating (e.g. VideoService")
operation	xsd: string	Yes	Parameter to use when performing rating (e.g. "streamVideo")

5.20.3.1 Example: get amount converted from volume (Informative)

5.20.3.1.1 Request

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MONTHLY-PLAN-001" HTTP/1.1
Accept: application/xml
Host: example.com
```

5.20.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:payment:1">
<chargingInformation>
  <description>100 Minutes converted to USD for endUserId=tel:+1-555-555-0100 according to contract="MONTHLY-PLAN-001"</description>
  <currency>USD</currency>
  <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

5.20.4 PUT

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

5.20.5 POST

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

5.20.6 DELETE

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

6. Fault definitions

6.1 Service Exceptions

None defined.

6.2 Policy Exceptions

The following Policy Exception codes are defined for the ParlayREST Payment API in addition to those in [3GPP 29.199-6].

6.2.1 POL0252: Refund request failed

Name	Description
MessageID	POL0252
Text	Refund request failed: %1.
Variables	%1 - textual description of failure. Value may be "OriginalServerReferenceCode is required in refund request", "Refund request amount exceeds original charge amount (charge amount included here)", "Refunds not supported by implementation".

6.2.2 POL0253: Payment operation refused by user

Name	Description
MessageID	POL0253
Text	Payment operation refused by user. %1
Variables	%1 – additional information about the refusal

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 1.1 History

Draft Versions: OMA-TS-ParlayREST_Payment-V1_1	12 Nov 2010	Many	Initial revision of V 1.1 and applied AI assigned to all editors as result of CONRR, as applicable to Payment: A124, A189.
	17 Nov 2010	Many	CR 555, CR 628, CR 611, CR 625R01
	08 Dec 2010	Many	CR 685R01, AI 276, AI 293, AI 307, plus editorials
	13 Dec 2010	2	Editorial fixes: cover page, Contents and references sorted
Candidate Version: OMA-TS-ParlayREST_Payment-V1_1	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.Payment Server

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

B.1.1 SCR for ParlayREST.Payment.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-TRANS-S-001-O	Support for access to all completed and pending transactions	5.4	PARLAYREST-PAY-TRANS-S-002-O
PARLAYREST-PAY-TRANS-S-002-O	Obtain all completed and pending transactions - GET	5.4.3	

B.1.2 SCR for ParlayREST.Payment.AmountCharge.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-AMNT-TRANS-S-001-M	Support for access to all amount charge and refund transactions for an end-user	5.5	
PARLAYREST-PAY-AMNT-TRANS-S-002-O	Obtain all amount charge and refund transactions for an end-user - GET	5.5.3	
PARLAYREST-PAY-AMNT-TRANS-S-003-M	Create a charge amount transaction for an end-user – POST (XML or JSON)	5.5.5	
PARLAYREST-PAY-AMNT-TRANS-S-004-O	Create a charge amount transaction for an end-user – POST (www-form-urlencoded)	C.1	
PARLAYREST-PAY-	Create a refund	5.5.5	

Item	Function	Reference	Requirement
AMNT-TRANS-S-005-M	amount transaction for an end-user - POST (XML or JSON)		
PARLAYREST-PAY-AMNT-TRANS-S-006-O	Create a refund amount transaction for an end-user – POST (www-form-urlencoded)	C.2	

B.1.3 SCR for ParlayREST.Payment.SplitAmount.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-SPLIT-TRANS-S-001-O	Support for access to all amount split charge and refund transactions for an end-user	5.6	PARLAYREST-PAY-SPLIT-TRANS-S-003-O
PARLAYREST-PAY-SPLIT-TRANS-S-002-O	Obtain all amount split charge transactions for an end-user - GET	5.6.3	
PARLAYREST-PAY-SPLIT-TRANS-S-003-O	Create a new amount split charge transaction for an end-user - POST	5.6.5	

B.1.4 SCR for ParlayREST.Payment.Individual.AmountCharge Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-AMNT-S-001-O	Support access to individual amount charge or refund for an end-user	5.7	PARLAYREST-PAY-IND-AMNT-S-002-O
PARLAYREST-PAY-IND-AMNT-S-002-O	Retrieve individual completed or pending amount transaction - GET	5.7.3	

B.1.5 SCR for ParlayREST.Payment.Individual.SplitAmount Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-SPLIT-S-001-O	Support access to individual split amount charge for an end-user	5.8	PARLAYREST-PAY-IND-SPLIT-S-002-O
PARLAYREST-PAY-IND-SPLIT-S-002-O	Retrieve individual completed or pending split amount transaction - GET	5.8.3	

B.1.6 SCR for ParlayREST.Payment.VolumeCharge.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-VOL-TRANS-S-001-O	Support for access to all volume charge and refund transactions for an end-user	5.9	PARLAYREST-PAY-VOL-TRANS-S-003-O AND PARLAYREST-PAY-VOL-TRANS-S-004-O
PARLAYREST-PAY-VOL-TRANS-S-002-O	Obtain all volume charge and refund transactions for an end-user - GET	5.9.3	
PARLAYREST-PAY-VOL-TRANS-S-003-O	Create a volume charge transaction for an end-user - POST	5.9.5	
PARLAYREST-PAY-VOL-TRANS-S-004-O	Create a volume refund transaction for an end-user - POST	5.9.5	

B.1.7 SCR for ParlayREST.Payment.Split.Volume.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-SPLIT-VOL-TRANS-S-001-O	Support for access to all volume split charge and refund transactions for an end-user	5.10	PARLAYREST-PAY-SPLIT-VOL-TRANS-S-003-O
PARLAYREST-PAY-SPLIT-VOL-TRANS-S-002-O	Obtain all volume split charge transactions for an end-user - GET	5.10.3	
PARLAYREST-PAY-SPLIT-VOL-TRANS-S-003-O	Create a new volume split charge transaction for an end-user - POST	5.10.5	

B.1.8 SCR for ParlayREST.Payment.Individual.VolumeCharge Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-VOL-S-001-O	Support access to individual volume charge or refund transaction for an end-user	5.11	PARLAYREST-PAY-IND-VOL-S-002-O
PARLAYREST-PAY-IND-VOL-S-002-O	Retrieve individual completed or pending transaction - GET	5.11.3	

B.1.9 SCR for ParlayREST.Payment.Individual.Split.VolumeCharge Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-SPLIT-VOL-S-001-O	Support access to individual split volume charge for an end-user	5.12	PARLAYREST-PAY-IND-SPLIT-VOL-S-002-O

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-SPLIT-VOL-S-002-O	Retrieve individual completed or pending split volume charge transaction - GET	5.12.3	

B.1.10 SCR for ParlayREST.Payment.Individual.Amount.VolumeCharge Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-AMNT-VOL-S-001-O	Support access to individual payment information based on volume transaction	5.13	PARLAYREST-PAY-IND-AMNT-VOL-S-002-O
PARLAYREST-PAY-IND-AMNT-VOL-S-002-O	Retrieve individual payment information based on volume transaction - GET	5.13.4	

B.1.11 SCR for ParlayREST.Payment.Individual.Amount.Split.VolumeCharge Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-001-O	Support access to individual payment information based on volume split transaction	5.14	PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-002-O
PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-002-O	Retrieve individual payment information based on volume split transaction - GET	5.14.4	

B.1.12 SCR for ParlayREST.Payment.Amount.Reserve.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-AMNT-RES-TRANS-S-001-M	Support for access to all completed and pending amount reservation transactions	5.15	
PARLAYREST-PAY-AMNT-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	5.15.4	
PARLAYREST-PAY-AMNT-RES-TRANS-S-003-M	Create a new amount reservation transaction for an end-user – POST (XML or JSON)	5.15.6	

Item	Function	Reference	Requirement
PARLAYREST-PAY-AMNT-RES-TRANS-S-004-O	Create a new amount reservation transaction for an end-user – POST (www-form-urlencoded)	C.3	

B.1.13 SCR for ParlayREST.Payment.Amount.Individual.Reserve.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-001-M	Support for access to individual completed and pending amount reservation transaction	5.16	
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	5.16.4	
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-003-M	Update an amount reservation transaction for an end-user – POST (XML or JSON)	5.16.6	
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-004-O	Update an amount reservation transaction for an end-user (reserve an additional amount) – POST (www-form-urlencoded)	C.4	
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-005-O	Update an amount reservation transaction for an end-user (charge to a reservation) – POST (www-form-urlencoded)	C.5	
PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-006-O	Update an amount reservation transaction for an end-user (release a reservation) – POST (www-form-urlencoded)	C.6	

B.1.14 SCR for ParlayREST.Payment.Volume.Reserve.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-VOL-RES-TRANS-S-001-O	Support for access to all volume reservation transactions	5.17	PARLAYREST-PAY-VOL-RES-TRANS-S-003-O
PARLAYREST-PAY-VOL-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	5.17.4	
PARLAYREST-PAY-VOL-RES-TRANS-S-	Create a new volume reservation transaction	5.17.6	

Item	Function	Reference	Requirement
003-O	for an end-user - POST		

B.1.15 SCR for ParlayREST.Payment.Volume.Individual.Reserve.Transactions Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-VOL-IND-RES-TRANS-S-001-O	Support for access to individual volume reservation transaction	5.18	PARLAYREST-PAY-VOL-IND-RES-TRANS-S-003-O
PARLAYREST-PAY-VOL-IND-RES-TRANS-S-002-O	Obtain individual transaction for an end-user - GET	5.18.4	
PARLAYREST-PAY-VOL-IND-RES-TRANS-S-003-O	Update a volume reservation transaction for an end-user - POST	5.18.6	

B.1.16 SCR for ParlayREST.Payment.Individual.Amount.Volume Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-IND-AMNT-VOL-S-001-O	Support for access to individual payment information based on volume reservation transaction	5.19	PARLAYREST-PAY-IND-AMNT-VOL-S-002-O
PARLAYREST-PAY-IND-AMNT-VOL-S-002-O	Obtain individual transaction for an end-user - GET	5.19.4	

B.1.17 SCR for ParlayREST.Payment.Amount.Convert Server

Item	Function	Reference	Requirement
PARLAYREST-PAY-AMNT-CONV-S-001-O	Support for access to amount converted from given volume	5.20	PARLAYREST-PAY-AMNT-CONV-S-002-O
PARLAYREST-PAY-AMNT-CONV-S-002-O	Return the amount resulting from converting the given volume for an end user - GET	5.20.4	

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

For selected operations, this section defines a format for Payment REST API requests where the body of the **request** is encoded using the *application/x-www-form-urlencoded* MIME type.

Note : only the request body is encoded as *application/x-www-form-urlencoded*, the response is still encoded as XML or JSON depending on the preference of the client and the capabilities of the server.

The following Payment REST operations are defined in this section:

- Charging an amount to an end user's account
- Refunding an amount to an end user' account
- Reserving a charge for an end user's account
- Adding/subtracting a charge to/from an existing reservation
- Charging to a previously made reservation
- Releasing funds left in a previously made reservation

C.1 Charge an Amount

This operation is used to charge a currency amount to an end-user account.

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '+' followed by the country code before the subscriber number; e.g.tel:+1-555-555-0100. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR.
transactionOperationStatus	xsd:string	No	Charged (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)
description	xsd:string	No	Information that appear on the bill
currency	xsd:string	Yes	The currency of the amount
amount	xsd:decimal	Yes	The amount to be charged. The amount to be charged appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied
referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a

			<p>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>
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies. How the consent is established is out of scope.
serviceID	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productID	xsd:string	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.1.1 Example

(Informative)

C.1.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

endUserId=tel%3A%2B1-555-555-0100&
transactionOperationStatus=Charged&
description=Test%20amount%20transaction%20%22Charged%22&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.1.1.2 Response

HTTP/1.1 201 Created

Content-Type: application/xml

Content-Length: nnnn

Date: Thu, 04 Jun 2009 02:51:59 GMT

Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123

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

```
<!-- AMOUNT CHARGE TRANSACTION -->
```

```
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
```

```
  <endUserId>tel:+1-555-555-0100</endUserId>
```

```
  <paymentAmount>
```

```
    <chargingInformation>
```

```
      <description>Test amount transaction "Charged" </description>
```

```
      <currency>USD</currency>
```

```
      <amount>10</amount>
```

```
      <code>TEST-012345</code>
```

```
    </chargingInformation>
```

```
    <totalAmountCharged>10</totalAmountCharged>
```

```
    <chargingMetaData>
```

```
      <onBehalfOf>Example Games Inc</onBehalfOf>
```

```
      <purchaseCategoryCode>Game</purchaseCategoryCode>
```

```
      <channel>WAP</channel>
```

```
      <taxAmount>0</taxAmount>
```

```
    </chargingMetaData>
```

```
  </paymentAmount>
```

```
  <transactionOperationStatus>Charged</transactionOperationStatus>
```

```
  <referenceCode>REF-12345</referenceCode>
```

```
</serverReferenceCode>ABC-123</serverReferenceCode>
```

```
  <clientCorrelator>54321</clientCorrelator>
```

```
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123</resourceURL>
```

```
</payment:amountTransaction>
```

C.2 Refund an Amount

This operation is used to refund a currency amount to an end-user account.

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '+' followed by the country code before the subscriber number; e.g.tel:+1-555-555-

			0100. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR.
transactionOperationStatus	TransactionOperationStatus	No	Refunded (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)
description	xsd:string	No	Information that appear on the bill
currency	xsd:string	Yes	The currency of the amount
amount	xsd:decimal	Yes	The amount to be refunded. The amount to be refunded appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied
referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
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>

			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.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceID	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productID	xsd:string	Yes	Combines with the service ID to uniquely identify the product being purchased.

C.2.1 Example

(Informative)

C.2.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

endUserId=tel%3A%2B1-555-555-0100&
transactionOperationStatus=Refunded&
description=Test%20amount%20transaction%20%22Refunded%22&
```



```

currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
originalServerReferenceCode=ABC123&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0

```

C.2.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans124

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>DEF-123</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans124</resourceURL>
</payment:amountTransaction>

```

C.3 Reserve an Amount

This operation is used to reserve a currency amount against an end-user account.

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '+' followed by the country code before the subscriber number; e.g.tel:+1-555-555-0100. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR.
transactionOperationStatus	TransactionOperationStatus	No	Reserved (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)
description	xsd:string	No	Information that appear on the bill
currency	xsd:string	Yes	The currency of the amount
amount	xsd:decimal	Yes	The amount to be reserved. The amount to be reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied
referenceSequence	xsd:string	No	Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc). The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server

			side).
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes
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>
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:decimal	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:string	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceID	xsd:string	Yes	The ID of the partner/merchant service being purchased.

productID	xsd:anyURI	Yes	Combines with the service ID to uniquely identify the product being purchased.
-----------	------------	-----	--------------------------------------------------------------------------------

C.3.1 Example

(Informative)

C.3.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

endUserId=tel%3A%2B1-555-555-0100&
transactionOperationStatus=Reserved&
description= Test%20amount%20transaction%20%22Reserved%22&
currency=USD&
amount=10&
referenceCode=TEST-012345&
referenceSequence=1&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.3.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction
  xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+1-555-555-0100</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  <chargingMetaData>
    <onBehalfOf>Example Games Inc</onBehalfOf>
```

```

<purchaseCategoryCode>Game</purchaseCategoryCode>
<channel>WAP</channel>
<taxAmount>0</taxAmount>
</chargingMetaData>
</paymentAmount>
<transactionOperationStatus>Reserved</transactionOperationStatus>
<referenceSequence>1</referenceSequence>
<referenceCode>TEST-012345</referenceCode>

<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

C.4 Reserve an Additional Amount

This operation is used to add a currency amount to an existing reservation.

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	Yes	Unique identifier for the end user's account. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '+' followed by the country code before the subscriber number; e.g. tel:+1-555-555-0100. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR.
transactionOperationStatus	TransactionOperationStatus	No	e.g. Charged, Reserved, etc (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings)
description	xsd:string	No	Information that appear on the bill
currency	xsd:string	Yes	The currency of the amount
amount	xsd:decimal	Yes	The amount to be reserved. The amount to be reserved appears

			either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes
referenceSequence	xsd:int	No	<p>Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc).</p> <p>The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).</p>
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being

			purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceID	xsd:anyURI	Yes	The ID of the partner/merchant service being purchased.
productID	xsd:anyURI	Yes	Combines with the service ID to uniquely identify the product being purchased.

C.4.1 Example

(Informative)

C.4.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
```

```
transactionOperationStatus=Reserved&
description=Test%20amount%20transaction%20%22Reserved%22&
amount=5&
referenceSequence=2
```

C.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Reserved"</description>
      <amount>5</amount>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>

```

C.5 Charge to a Reservation

This operation is used to charge against an existing reservation.

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	Yes	Unique identifier for the end user's account. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '+' followed by the country code before the subscriber number; e.g.tel:+1-555-555-0100. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR.
transactionOperationStatus	TransactionOperationStatus	No	Charged (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)
description	xsd:string	No	Information that appear on the bill
currency	xsd:string	Yes	The currency of the amount
amount	xsd:decimal	Yes	The amount to be charged. The amount to be charged appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied

referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc). The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceID	xsd:anyURI	Yes	The ID of the partner/merchant service being purchased.
productID	xsd:anyURI	Yes	Combines with the service ID to uniquely identify the product being purchased.

C.5.1 Example

(Informative)

C.5.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: nnnn
```

```
transactionOperationStatus=Charged&
```

```
description=Test%20amount%20transaction%20in%22Refunded%22&
```

```
amount=5&
```

```
referenceCode=REF-12345&
```

```
referenceSequence=3&
```

```
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.5.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged" </description>
      <amount>5</amount>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>10</amountReserved>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

C.6 Release a Reservation

This operation is used to return the funds in an existing reservation.

The request parameters are as follows:

Name	Type/Values	Optional	Description
transactionOperationStatus	TransactionOperationStatus	No	Released (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)
description	xsd:string	No	Information that appear on the bill

code	xsd:string	Yes	Charging code referencing a contract under which the charge is applied
referenceSequence	xsd:int	No	<p>Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc).</p> <p>The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server side).</p>
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes
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>
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.

mandateID	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceID	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productID	xsd:string	Yes	Combines with the service ID to uniquely identify the product being purchased.

C.6.1 Example

(Informative)

C.6.1.1 Request

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

transactionOperationStatus=Released&
description=Test%20amount%20transaction%20%22Released%22&
code=TEST012345&
referenceSequence=4
```

C.6.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction
  xmlns:payment="urn:oma:xml:rest:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Released" </description>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Released</transactionOperationStatus>
  <referenceSequence>4</referenceSequence>
  <resourceURL>http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

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 Get all transactions (section 5.4.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountReservationTransaction": [
    {
      "clientCorrelator": "55555",
      "endUserId": "tel:+1-555-555-0100",
      "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
          "amount": "15",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount reservation transaction \"Charged\""
        }
      },
      "totalAmountCharged": "25"
    },
    {
      "referenceCode": "REF-12345",
      "referenceSequence": "2",
      "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans128",
      "serverReferenceCode": "PQR-123",
      "transactionOperationStatus": "Charged"
    }
  ],
  {
    "clientCorrelator": "55556",
    "endUserId": "tel:+1-555-555-0100",
    "paymentAmount": {
      "amountReserved": "10",
      "chargingInformation": {
```

```

        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \"Reserved\""
    },
    "totalAmountCharged": "0"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans998",
"transactionOperationStatus": "Reserved"
}
],
"amountSplitTransaction": {
    "clientCorrelator": "54323",
    "endUserShare": [
        {
            "endUserId": "tel:+1-555-555-0100",
            "percent": "20"
        },
        {
            "endUserId": "tel:+1-555-555-0101",
            "percent": "80"
        }
    ],
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\""
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans789",
    "serverReferenceCode": "ABC-789",
    "transactionOperationStatus": "Charged"
},
"amountTransaction": [
    {
        "clientCorrelator": "54321",
        "endUserId": "tel:+1-555-555-0100",
        "paymentAmount": {
            "chargingInformation": {
                "amount": "10",
                "code": "TEST-012345",
                "currency": "USD",
                "description": "Test amount transaction \"Charged\""
            },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123",
        "serverReferenceCode": "ABC-123",

```

```

    "transactionOperationStatus": "Charged"
  },
  {
    "clientCorrelator": "54322",
    "endUserId": "tel:+1-555-555-0100",
    "originalServerReferenceCode": "ABC-123",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount transaction \"Refunded\""
      },
      "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans456",
    "serverReferenceCode": "ABC-456",
    "transactionOperationStatus": "Refunded"
  }
],
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions",
"volumeReservationTransaction": [
  {
    "clientCorrelator": "66666",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Reserved\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "0",
      "volume": "10",
      "volumeReserved": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999",
    "transactionOperationStatus": "Reserved"
  },
  {
    "clientCorrelator": "66667",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Charged\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "25",
      "volume": "10",
      "volumeReserved": "0"
    },
    "referenceCode": "REF-12345",
  }
]

```

```

    "referenceSequence": "4",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/volumeReservation/trans127",
    "serverReferenceCode": "MNO-123",
    "transactionOperationStatus": "Charged"
  }
],
"volumeSplitTransaction": {
  "clientCorrelator": "54325",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "20"
    },
    {
      "endUserId": "tel:+1-555-555-0101",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans126",
  "serverReferenceCode": "JKL-123",
  "transactionOperationStatus": "Charged"
},
"volumeTransaction": [
  {
    "clientCorrelator": "55551",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124",
    "serverReferenceCode": "DEF-123",
    "transactionOperationStatus": "Charged"
  },
  {
    "clientCorrelator": "54324",
    "endUserId": "tel:+1-555-555-0100",
    "originalServerReferenceCode": "DEF-123",
    "paymentVolume": {

```



```

    "billingText": "Test volume transaction \"Refunded\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeRefunded": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans125",
  "serverReferenceCode": "GHI-123",
  "transactionOperationStatus": "Refunded"
}
]
}}

```

D.2 Request with invalid endUserId (section 5.4.3.2)

Request:

```

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

```

Response:

```

HTTP/1.1 404 Not Found
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {
  "link": {
    "href": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions",
    "rel": "PaymentTransactionList"
  },
  "serviceException": {
    "messageId": "SVC0004",
    "text": "Invalid input value. The address %1 does not exist.",
    "variables": "tel:+1-555-555-0100"
  }
}
}}

```

D.3 Get all amount transactions (section 5.5.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "paymentTransactionList": {
    "amountTransaction": [
      {
        "clientCorrelator": "54321",
        "endUserId": "tel:+1-555-555-0100",
        "paymentAmount": {
          "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\""
          },
          "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
      },
      {
        "clientCorrelator": "54321",
        "endUserId": "tel:+1-555-555-0100",
        "paymentAmount": {
          "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\""
          },
          "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans124",
        "serverReferenceCode": "DEF-123",
        "transactionOperationStatus": "Charged"
      },
      {
        "clientCorrelator": "54330",
        "endUserId": "tel:+1-555-555-0100",
        "originalServerReferenceCode": "DEF-123",
        "paymentAmount": {
          "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount transaction \"Refunded\""
          },
          "totalAmountRefunded": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans129",
      }
    ]
  }
}
```

```

    "serverReferenceCode": "WXY-123",
    "transactionOperationStatus": "Refunded"
  }
],
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount"
}}

```

D.4 Create charge amount (section 5.5.5.1)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

```

```

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction \"Charged\""
  }},
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\""
    }
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}

```

D.5 Create refund amount (section 5.5.5.2)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "originalServerReferenceCode": "ABC-123",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction \"Refunded\""
  }},
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Refunded"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans130

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "originalServerReferenceCode": "ABC-123",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Refunded\""
    }
  },
  "totalAmountRefunded": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans130",
"serverReferenceCode": "XYZ-123",
"transactionOperationStatus": "Refunded"
}}
```

D.6 Client retries POST with same clientCorrelator (section 5.5.5.3)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMTLocation: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\" "
    }
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}
```

D.7 Unsuccessful charge request because of denial/refusal by back-end system (section 5.5.5.4)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC0270",
  "text": "Charging operation failed, the charge was not applied."
}}}
```

D.8 Get all amount split transactions (section 5.6.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountSplitTransaction": [
    {
      "clientCorrelator": "55552",
      "endUserShare": [
        {
          "endUserId": "tel:+1-555-555-0100",
```

```

        "percent": "30"
    },
    {
        "endUserId": "tel:+1-555-555-0101",
        "percent": "70"
    }
],
"paymentAmount": {
    "chargingInformation": {
        "amount": "20",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction \"Charged\""
    },
    "totalAmountCharged": "20"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
},
{
    "clientCorrelator": "55553",
    "endUserShare": [
        {
            "endUserId": "tel:+1-555-555-0100",
            "percent": "20"
        },
        {
            "endUserId": "tel:+1-555-555-0101",
            "percent": "80"
        }
    ],
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012346",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\" "
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12346",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans124",
    "serverReferenceCode": "DEF-123",
    "transactionOperationStatus": "Charged"
}
],
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit"
}}

```

D.9 Create split charge amount (section 5.6.5.1)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountSplitTransaction": {
  "clientCorrelator": "54431",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "30"
    },
    {
      "endUserId": "tel:+1-555-555-0101",
      "percent": "70"
    }
  ],
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123

{"amountSplitTransaction": {
  "clientCorrelator": "54431",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "30"
    },
    {
      "endUserId": "tel:+1-555-555-0101",
      "percent": "70"
    }
  ],
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
```



```

    "currency": "USD",
    "description": "Test amount transaction \"Charged\" "
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}

```

D.10 Get amount charge (section 5.7.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123?resFormat=JSON HTTP/1.1
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\" "
    }
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amount/trans123",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}

```

D.11 Get amount split charge (section 5.8.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123
HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountSplitTransaction": {
  "clientCorrelator": "54321",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "30"
    },
    {
      "endUserId": "tel:+1-555-555-0101",
      "percent": "70"
    }
  ],
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\" "
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountSplit/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}

```

D.12 Get all volume charge and refund transactions (section 5.9.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume",
  "volumeTransaction": [
    {

```

```

    "clientCorrelator": "55551",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\" ",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
  },
  {
    "clientCorrelator": "55552",
    "endUserId": "tel:+1-555-555-0100",
    "originalServerReferenceCode": "ABC-123",
    "paymentVolume": {
      "billingText": "Test volume transaction \"Refunded\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeRefunded": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124",
    "serverReferenceCode": "DEF-123",
    "transactionOperationStatus": "Refunded"
  }
]
}}

```

D.13 Create charge volume, returning a representation of created resource (section 5.9.5.1)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json

{"volumeTransaction": {
  "clientCorrelator": "55551",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {

```

```

    "name": "unit",
    "value": "minutes"
  },
  "volume": "10"
},
"referenceCode": "REF-12345",
"transactionOperationStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeTransaction": {
  "clientCorrelator": "55551",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}

```

D.14 Create charge volume, returning the location of created resource (section 5.9.5.2)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json

{"volumeTransaction": {
  "clientCorrelator": "55551",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",

```

```

    "value": "minutes"
  },
  "volume": "10"
},
"referenceCode": "REF-12345",
"transactionOperationStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123"}}

```

D.15 Create refund volume (section 5.9.5.3)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml

{"volumeTransaction": {
  "clientCorrelator": "55552",
  "endUserId": "tel:+1-555-555-0100",
  "originalServerReferenceCode": "ABC-123",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Refunded\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Refunded"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124

{"volumeTransaction": {

```

```

"clientCorrelator": "55552",
"endUserId": "tel:+1-555-555-0100",
"originalServerReferenceCode": "ABC-123",
"paymentVolume": {
  "billingText": "Test volume transaction \"Refunded\"",
  "ratingParameter": {
    "name": "unit",
    "value": "minutes"
  },
  "totalVolumeRefunded": "10",
  "volume": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans124",
"serverReferenceCode": "DEF-123",
"transactionOperationStatus": "Refunded"
}}

```

D.16 Get all volume split charge transactions (section 5.10.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit
HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit",
  "volumeSplitTransaction": {
    "clientCorrelator": "55553",
    "endUserShare": [
      {
        "endUserId": "tel:+1-555-555-0100",
        "percent": "20"
      },
      {
        "endUserId": "tel:+1-555-555-0101",
        "percent": "80"
      }
    ],
    "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      }
    }
  }
}

```

```

    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}
}}

```

D.17 Create volume split charge (section 5.10.5.1)

Request:

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit HTTP/1.1

Accept: application/json

Host: example.com

Content-Type: application/json

Content-Length: nnnn

```

{"volumeSplitTransaction": {
  "clientCorrelator": "55553",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "20"
    },
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}

```

Response:

HTTP/1.1 201 Created

Content-Type: application/json

Content-Length: nnnn

Date: Thu, 04 Jun 2009 02:51:59 GMT

Location: <http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123>

```
{
  "volumeSplitTransaction": {
    "clientCorrelator": "55553",
    "endUserShare": [
      {
        "endUserId": "tel:+1-555-555-0100",
        "percent": "20"
      },
      {
        "endUserId": "tel:+1-555-555-0101",
        "percent": "80"
      }
    ],
    "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\" ",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
  }
}
```

D.18 Get volume charge (section 5.11.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
```



```

    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volume/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}

```

D.19 Get volume split charge (section 5.12.3.1)

Request:

```

GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123 HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeSplitTransaction": {
  "clientCorrelator": "55556",
  "endUserShare": [
    {
      "endUserId": "tel:+1-555-555-0100",
      "percent": "20"
    },
    {
      "endUserId": "tel:+1-555-555-0101",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}

```

D.20 Get amount for volume charge (section 5.13.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/volume/{transactionId}/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+1-555-555-0100"
}}}
```

D.21 Get amount for volume split charge (section 5.14.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeSplit/trans999/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "10 Minutes converted to USD for transaction=trans999"
}}}
```

D.22 Get all amount reservation transactions (section 5.15.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation HTTP/1.1
Accept: application/json
```

Host: example.com

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountReservationTransaction": [
    {
      "clientCorrelator": "55555",
      "endUserId": "tel:+1-555-555-0100",
      "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
          "amount": "15",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount reservation transaction \"Charged\" "
        }
      },
      "totalAmountCharged": "25"
    },
    {
      "referenceCode": "REF-12345",
      "referenceSequence": "2",
      "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans123",
      "serverReferenceCode": "ABC-123",
      "transactionOperationStatus": "Charged"
    },
    {
      "clientCorrelator": "55556",
      "endUserId": "tel:+1-555-555-0100",
      "paymentAmount": {
        "amountReserved": "10",
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount reservation transaction \"Reserved\""
        }
      },
      "totalAmountCharged": "0"
    },
    {
      "referenceSequence": "1",
      "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
      "transactionOperationStatus": "Reserved"
    },
    {
      "clientCorrelator": "55557",
      "endUserId": "tel:+1-555-555-0100",
      "paymentAmount": {
        "amountReserved": "15",
        "chargingInformation": {

```

```

        "amount": "15",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \"Reserved\""
    },
    "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans998",
"transactionOperationStatus": "Reserved"
},
{
    "clientCorrelator": "55557",
    "endUserId": "tel:+1-555-555-0100",
    "paymentAmount": {
        "amountReserved": "25",
        "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Reserved\""
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "3",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans997",
    "transactionOperationStatus": "Reserved"
}
],
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation"
}}

```

D.23 Create reserve amount (section 5.15.5.1)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

```

```

{"amountReservationTransaction": {
    "clientCorrelator": "55555",
    "endUserId": "tel:+1-555-555-0100",
    "paymentAmount": {"chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \"Reserved\""
    }
}},

```

```
"referenceSequence": "1",
"transactionOperationStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999
```

```
{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Reserved\""
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "1",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Reserved"
}}
```

D.24 Get amount reservation (section 5.16.3.1)

Request:

```
GET /exampleAPI/1/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "10",
```

```

    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Reserved\""
  },
  "totalAmountCharged": "0"
},
"referenceSequence": "1",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
"transactionOperationStatus": "Reserved"
}}

```

D.25 Charge amount for amount reservation (section 5.16.5.1)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Charged\" "
    }
  }
},

```

```

    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Charged"
}}

```

D.26 Repeat charge request with same referenceSequence for an amount reservation (section 5.16.5.2)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Charged\" "
    }
  },
  "totalAmountCharged": "10"
}

```

```

},
"referenceCode": "REF-12345",
"referenceSequence": "2",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999",
"transactionOperationStatus": "Charged"
}}

```

D.27 Release amount reservation (section 5.16.5.3)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "code": "TEST012345",
    "description": "Test amount reservation transaction \\"Released\\""
  }},
  "referenceSequence": "2",
  "transactionOperationStatus": "Released"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55556",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \\"Released\\""
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-
0100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Released"
}}

```


D.28 Charge partial amount for amount reservation (section 5.16.5.4)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "5",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Charged\" "
  }},
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55557",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "5",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Charged\" "
    }
  },
  "totalAmountCharged": "5"
},
"referenceCode": "REF-12345",
"referenceSequence": "2",
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
"transactionOperationStatus": "Charged"
}}
```

D.29 Release remaining amount reservation (section 5.16.5.5)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "code": "TEST012345",
    "description": "Test amount reservation transaction \"Released\""
  }},
  "referenceSequence": "3",
  "transactionOperationStatus": "Released"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55558",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Released\""
    },
    "totalAmountCharged": "5"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "3",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Released"
}}
```

D.30 Reserve additional amount for amount reservation (section 5.16.5.6)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
```

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

```
{"amountReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {"chargingInformation": {
    "amount": "5",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Reserved\""
  }},
  "referenceSequence": "2",
  "transactionOperationStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{"amountReservationTransaction": {
  "clientCorrelator": "55559",
  "endUserId": "tel:+1-555-555-0100",
  "paymentAmount": {
    "amountReserved": "15",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Reserved\""
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Reserved"
}}
```

D.31 Unsuccessful charge amount for amount reservation because of denial/refusal by back-end system (section 5.16.5.7)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
```

Content-Type: application/json
Content-Length: nnnn

```
{
  "amountReservationTransaction": {
    "endUserId": "tel:+1-555-555-0100",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \\\"Charged\\\" "
      }
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "transactionOperationStatus": "Charged"
  }
}
```

Response:

HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "requestError": {
    "link": {
      "href": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/amountReservation/trans999",
      "rel": "AmountReservationTransaction"
    },
    "serviceException": {
      "messageId": "SVC0270",
      "text": "Charging operation failed, the charge was not applied."
    }
  }
}
```

D.32 Get all volume reservation transactions (section 5.17.3.1)

Request:

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

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "paymentTransactionList": {
```

```

"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation",
"volumeReservationTransaction": [
  {
    "clientCorrelator": "66666",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Reserved\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "0",
      "volume": "10",
      "volumeReserved": "10"
    },
    "referenceSequence": "1",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999",
    "transactionOperationStatus": "Reserved"
  },
  {
    "clientCorrelator": "66667",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Reserved\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "15",
      "volume": "10",
      "volumeReserved": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "3",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans998",
    "transactionOperationStatus": "Reserved"
  },
  {
    "clientCorrelator": "66668",
    "endUserId": "tel:+1-555-555-0100",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Charged\" ",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "25",
      "volume": "10",
      "volumeReserved": "0"
    },
    "referenceCode": "REF-22345",
    "referenceSequence": "4",
    "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans123",

```

```

    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
  }
]
}}

```

D.33 Create reserve volume (section 5.17.5.1)

Request:

```

POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceSequence": "1",
  "transactionOperationStatus": "Reserved"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "0",
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceSequence": "1",

```

```
"resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999",
"transactionOperationStatus": "Reserved"
}}
```

D.34 Create reserve volume with invalid endUserID (section 5.17.5.2)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
  "clientCorrelator": "66667",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceSequence": "1",
  "transactionOperationStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC0004",
  "text": "Invalid input value. The address %1 does not exist.",
  "variables": "tel:+1-555-555-0100"
}}}
}}
```

D.35 Get volume reservation (section 5.18.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/xml
```

Host: example.com

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "0",
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999",
  "transactionOperationStatus": "Reserved"
}}
```

D.36 Charge volume (section 5.18.5.1)

Request:

```
POST /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}
```


Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+1-555-555-0100",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10",
    "volumeReserved": "0"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999",
  "transactionOperationStatus": "Charged"
}}
```

D.37 Get amount for volume reservation (section 5.19.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/transactions/volumeReservation/trans999/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+1-555-555-0100"
}}}
```

D.38 Get amount converted from volume (section 5.20.3.1)

Request:

```
GET /exampleAPI/1/payment/tel%3A%2B1-555-555-0100/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MONTHLY-PLAN-001"&currency="USD" HTTP 1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "100 Minutes converted to USD for endUserId=tel:+1-555-555-0100 according to contract="MONTHLY-PLAN-001"
}}}
```

Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/operations and Parlay X equivalent operations.

ParlayREST Resource	ParlayREST method	ParlayREST section reference	Parlay X equivalent operations
All amount charge and refund transactions for an end user	POST	5.5.5	chargeAmount and refundAmount
All amount split charge transactions for an end user	POST	5.6.5	chargeSplitAmount
All volume charge and refund transactions for an end user	POST	5.9.5	chargeVolume and refundVolume
All volume split charge transactions for an end user	POST	5.10.5	chargeSplitVolume
All amount reservation transactions for an end user	POST	5.15.5	reserveAmount
Individual amount reservation transaction for an end user	POST	5.16.5	reserveAdditionalAmount, chargeReservation and releaseReservation
All volume reservation transactions for an end user	POST	5.17.5	reserveVolume
Individual volume reservation transaction for an end user	POST	5.18.5	reserveAdditionalVolume, chargeReservation and releaseReservation
Amount converted from given volume	GET	5.20.3	getAmount

Table 1: Parlay X operations mapping

Appendix F. Mapping of ParlayREST Reservation Transactions to the Charging Enabler (Informative)

ParlayREST contains functionality for payment operations which has been designed such that it can be easily mapped to Parlay X operations. The OMA Charging Enabler provides details for Event and Session based Charging. A mapping of ParlayREST to the OMA Charging Enabler is straightforward in the case of Event based Charging. For the case of Session based Charging, this appendix elaborates on the details of the mapping because the underlying assumptions are different in the OMA Charging Enabler and Parlay X / ParlayREST.

ParlayREST contains functionality for Reservation-based payment operations (sections 5.15, 5.16, 5.17, 5.18). The specification of Session based Charging in the OMA Charging Enabler defines the use of the Reserve Units, ReserveAndDebit Units and Debit Units operations and the respective messages, i.e. Charging Request and Charging Response message, in [CHRG TS ONLINE].

The section below describes the mapping of the ParlayREST Reservation Transactions to the Reserve Units and Debit Units operations can be achieved.

F.1 Charging Mechanism

Charging uses the Reserve Units, ReserveAndDebit Units and Debit Units operations and allows to open a charging session, which starts by reserving units (amount or volume), then continuously doing debiting of the used units, as well as reservation of new units, and finally closing the charging session [CHRG TS ONLINE].

The three phases of a Charging session are as follows:

1. In a first step (first interrogation), a Reserve Units operation is performed. This maps 1:1 to the reserving operation “nil → reserved”. The application acting as a Charging Enabler User requests a number of units (money, data volume, usage time etc.), called RSU (Requested Service Units), from the Charging Enabler. At the end of this step, the Charging Enabler has deducted a certain number of units from the user’s account, and has granted them to the Charging Enabler User for consumption. These units are named GSU (Granted Service Units).
2. In subsequent steps (intermediate interrogation), the ReserveAndDebit Units operation is used by the application to debit consumed units and to make new reservations in one operation. Such ReserveAndDebit Units request includes the number of Used Service Units (USU) for debiting and a number of Requested Service Units (RSU) for subsequent consumption. Note that all GSU from a previous step are rendered invalid at this point in time. The Charging Enabler debits the USU to the user’s account, and returns a new GSU to the application for consumption. This step can be performed multiple times.
3. In a last step (final interrogation), the last debiting of Used Service Units (USU) is invoked by the application with the Debit Units operation. The Charging Enabler debits the USUs, puts the remaining GSUs back into the user’s account and closes the session.

Compared to the Parlay X Payment API, in the Charging Enabler there is a tighter coupling of reservation and debiting. Parlay X has separate methods for doing an initial reservation (ReserveAmount / ReserveVolume), reserving additional units (ReserveAdditionalAmount / ReserveAdditionalVolume), debiting used units (ChargeReservation) and releasing the session (ReleaseReservation). In ParlayREST, these are reflected by the separate POST requests. In the Charging Enabler, debiting and reserving, as well as debiting and releasing, are coupled.

F.2 Detailed Mapping of ParlayREST to Charging Enabler

The previous section has outlined the basic flow in Session Charging, and the different approaches in ParlayX / ParlayREST and the OMA Charging Enabler. The OMA Charging Enabler maps the Reserve Units, ReserveAndDebit Units and Debit Units operations to Charging Request and Charging Response message pairs.

This section describes a possible realization of the transaction operation state transitions in ParlayREST (see section 5.2.13) based on those Charging Request and Charging Response message pairs.

In order to be able to use a unified pattern for amount and volume charging, the following conventions are used below: The variable *amountVolumeReserved* corresponds to the ParlayREST elements *amountReserved* and *volumeReserved*. The variable *totalAmountVolumeCharged* corresponds to the ParlayREST elements *totalAmountCharged* and *totalVolumeCharged*. The variable *amountVolume* corresponds to the ParlayREST elements *paymentAmount/amount* and *paymentVolume/volume*. The parameters GSU, RSU and USU are as defined above.

In the following mappings the USU parameter is set to zero wherever applied indicate that no units has to be debited. It is also taken into account that the credit expires at each intermediate interrogation in the Charging Enabler, but that reservations are cumulative in Parlay X and therefore in ParlayREST.

1) initial reserving as part of ParlayREST resource creation

Remark: this operation has a direct correspondence in the Reserve Units operation

```
RSU=amountVolume
send Charging Request(INITIAL, RSU) and receive Charging Response(INITIAL, GSU)
amountVolumeReserved=GSU
```

2) charging as ParlayREST resource update

Remark: here, the RSU needs to reflect the previous reservation minus what is requested to be debited

```
USU=amountVolume
RSU=max[amountVolumeReserved-amountVolume,0]
send Charging Request(UPDATE, USU, RSU) and receive Charging Response(UPDATE, GSU)
totalAmountVolumeCharged=totalAmountVolumeCharged+amountVolume
amountVolumeReserved=GSU
```

3) reserving as ParlayREST resource update

Remark: here, the RSU needs to reflect the previous reservation plus what is requested to be reserved. USU set to zero as there is no debit here.

```
RSU=amountVolumeReserved+amountVolume
USU=0
send Charging Request(UPDATE, USU, RSU) and receive Charging Response(UPDATE, GSU)
amountVolumeReserved=GSU
```

4) releasing as ParlayREST resource update

Remark: the USU is set to zero as there is no debit here.

```
USU=0
send Charging Request(TERMINATION, USU) and receive Charging Response(TERMINATION, USU)
amountVolumeReserved=0
```

Note that the sequence (2), (3) can be called 0 or more times.

Note: In case of insufficient credit for the steps (1) and (3) one of the following cases depending on operator policy and/or particular deployment may occur: The server returns an actual reservation that is smaller than the requested one or throws an exception. A client needs to be prepared to handle both cases.