

RESTful bindings for Parlay X Web Services - Payment

Candidate Version 1.0 – 27 Apr 2010

Open Mobile Alliance

OMA-TS-ParlayREST_Payment-V1_0-20100427-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.

© 2010 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 | 23 |
| 5.2.1 | Type: PaymentTransactionList | 23 |
| 5.2.2 | Type: AmountTransaction | 24 |
| 5.2.3 | Type: AmountSplitTransaction..... | 25 |
| 5.2.4 | Type: VolumeTransaction..... | 25 |
| 5.2.5 | Type: VolumeSplitTransaction..... | 26 |
| 5.2.6 | Type: AmountReservationTransaction | 27 |
| 5.2.7 | Type: VolumeReservationTransaction..... | 28 |
| 5.2.8 | Type: PaymentAmount | 29 |
| 5.2.9 | Type: PaymentVolume | 30 |
| 5.2.10 | Type: ChargingMetaData..... | 30 |
| 5.2.11 | Type: EndUserShare | 31 |
| 5.2.12 | Type: RatingParameter | 31 |
| 5.2.13 | Enumeration: TransactionStatus | 32 |
| 5.2.14 | Values of the Link “rel” attribute..... | 32 |
| 5.3 | SEQUENCE DIAGRAMS..... | 33 |
| 5.3.1 | Amount charge and refund transaction | 33 |
| 5.3.2 | Amount split charge transaction | 34 |
| 5.3.3 | Volume charge and refund transaction | 34 |
| 5.3.4 | Volume split charge transaction..... | 35 |
| 5.3.5 | Amount reservation and release transaction | 36 |
| 5.3.6 | Volume reservation and release 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 | <i>HTTP Response Codes</i> | 41 |
| 5.4.2.2 | <i>Exception fault codes</i> | 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 endUserId (Informative) | 45 |
| 5.4.3.2.1 | Request | 45 |
| 5.4.3.2.2 | Response | 45 |
| 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 | 46 |

- 5.5.2.1 *HTTP Response Codes*..... 46
- 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)* 50
 - 5.5.5.2.1 Request..... 50
 - 5.5.5.2.2 Response..... 50
 - 5.5.5.3 *Example 3: client retries POST with same clientCorrelator (Informative)*..... 51

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 [TS_Common] section 5.6.1 for other possible cases. 51

 - 5.5.5.3.1 Request..... 51
 - 5.5.5.3.2 Response..... 51
- 5.5.6 **DELETE**..... 52
- 5.6 RESOURCE: ALL AMOUNT SPLIT CHARGE TRANSACTIONS FOR AN END USER 52**
 - 5.6.1 **Request URI variables** 52
 - 5.6.2 **Response Codes** 52
 - 5.6.2.1 *HTTP Response Codes*..... 52
 - 5.6.2.2 *Exception fault codes* 52
 - 5.6.3 **GET** 52
 - 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** 54
 - 5.6.5 **POST** 54
 - 5.6.5.1 *Example: create split charge amount (Informative)*..... 54
 - 5.6.5.1.1 Request..... 54
 - 5.6.5.1.2 Response..... 55
 - 5.6.6 **DELETE**..... 55
- 5.7 RESOURCE: INDIVIDUAL AMOUNT CHARGE OR REFUND TRANSACTION FOR AN END USER..... 56**
 - 5.7.1 Request URI variables 56
 - 5.7.2 Response Codes 56
 - 5.7.2.1 *HTTP Response Codes*..... 56
 - 5.7.2.2 *Exception fault codes* 56
 - 5.7.3 **GET** 56
 - 5.7.3.1 *Example: get amount charge (Informative)* 56
 - 5.7.3.1.1 Request..... 56
 - 5.7.3.1.2 Response..... 57
 - 5.7.4 **PUT** 57
 - 5.7.4.1 *Example: update amount transaction (Informative)* 57
 - 5.7.4.1.1 Request..... 57
 - 5.7.4.1.2 Response..... 58
 - 5.7.5 **POST**..... 58
 - 5.7.6 **DELETE**..... 58
- 5.8 RESOURCE: INDIVIDUAL AMOUNT SPLIT CHARGE TRANSACTION FOR AN END USER 59**
 - 5.8.1 **Request URI variables** 59
 - 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..... 60
 - 5.8.4 **PUT**..... 60
 - 5.8.4.1 *Example: update split charge amount transaction (Informative)*..... 60

- 5.8.4.1.1 Request..... 60
- 5.8.4.1.2 Response..... 61
- 5.8.5 POST..... 62
- 5.8.6 DELETE..... 62
- 5.9 RESOURCE: ALL VOLUME CHARGE AND REFUND TRANSACTIONS FOR AN END USER..... 62**
 - 5.9.1 Request URI variables 62
 - 5.9.2 Response Codes 62
 - 5.9.2.1 HTTP Response Codes..... 62
 - 5.9.2.2 Exception fault codes..... 63
 - 5.9.3 GET..... 63
 - 5.9.3.1 Example: get all volume charge and refund transactions (Informative)..... 63
 - 5.9.3.1.1 Request..... 63
 - 5.9.3.1.2 Response..... 63
 - 5.9.4 PUT 64
 - 5.9.5 POST 64
 - 5.9.5.1 Example 1: create charge volume, returning a representation of created resource (Informative) 64
 - 5.9.5.1.1 Request..... 64
 - 5.9.5.1.2 Response..... 65
 - 5.9.5.2 Example 2: create charge volume, returning the location of created resource (Informative)..... 65
 - 5.9.5.2.1 Request..... 65
 - 5.9.5.2.2 Response..... 66
 - 5.9.5.2.3 Example 3: create refund volume (Informative) 66**
 - 5.9.5.2.4 Request..... 66
 - 5.9.5.2.5 Response..... 66
 - 5.9.6 DELETE..... 67

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]...... 67
- 5.10 RESOURCE: ALL VOLUME SPLIT CHARGE TRANSACTIONS FOR AN END USER 67**
 - 5.10.1 Request URI variables 67
 - 5.10.2 Response Codes 68
 - 5.10.2.1 HTTP Response Codes 68
 - 5.10.2.2 Exception fault codes..... 68
 - 5.10.3 GET..... 68
 - 5.10.3.1 Example: get all volume split charge transactions (Informative)..... 68
 - 5.10.3.1.1 Request..... 68
 - 5.10.3.1.2 Response..... 68
 - 5.10.4 PUT 69
 - 5.10.5 POST 69
 - 5.10.5.1 Example: create volume split charge (Informative)..... 69
 - 5.10.5.1.1 Request..... 69
 - 5.10.5.1.2 Response..... 70
 - 5.10.6 DELETE..... 70
- 5.11 RESOURCE: INDIVIDUAL VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER..... 70**
 - 5.11.1 Request URI variables 71
 - 5.11.2 Response Codes 71
 - 5.11.2.1 HTTP Response Codes 71
 - 5.11.2.2 Exception fault codes..... 71
 - 5.11.3 GET 71
 - 5.11.3.1 Example: get volume charge (Informative) 71
 - 5.11.3.1.1 Request..... 71
 - 5.11.3.1.2 Response..... 71
 - 5.11.4 PUT..... 72
 - 5.11.4.1 Example: update volume charge transaction (Informative)..... 72
 - 5.11.4.1.1 Request..... 72
 - 5.11.4.1.2 Response..... 73
 - 5.11.5 POST..... 73
 - 5.11.6 DELETE..... 73
- 5.12 RESOURCE: INDIVIDUAL VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER 73**
 - 5.12.1 Request URI variables 74
 - 5.12.2 Response Codes 74
 - 5.12.2.1 HTTP Response Codes 74

- 5.12.2.2 *Exception fault codes*..... 74
- 5.12.3 **GET** 74
 - 5.12.3.1 *Example: get volume split charge (Informative)*..... 74
 - 5.12.3.1.1 Request..... 74
 - 5.12.3.1.2 Response..... 74
- 5.12.4 **PUT** 75
 - 5.12.4.1 *Example: update volume split charge transaction (Informative)*..... 75
 - 5.12.4.1.1 Request..... 75
 - 5.12.4.1.2 Response..... 76
- 5.12.5 **POST**..... 77
- 5.12.6 **DELETE**..... 77
- 5.13 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER..... 77**
 - 5.13.1 **Request URI variables** 77
 - 5.13.2 **Response Codes** 77
 - 5.13.2.1 *HTTP Response Codes* 77
 - 5.13.2.2 *Exception fault codes*..... 77
 - 5.13.3 **GET** 78
 - 5.13.3.1 *Example: get amount for volume charge (Informative)* 78
 - 5.13.3.1.1 Request..... 78
 - 5.13.3.1.2 Response..... 78
 - 5.13.4 **PUT** 78
 - 5.13.5 **POST**..... 78
 - 5.13.6 **DELETE**..... 78
- 5.14 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER..... 79**
 - 5.14.1 **Request URI variables** 79
 - 5.14.2 **Response Codes** 79
 - 5.14.2.1 *HTTP Response Codes* 79
 - 5.14.2.2 *Exception fault codes*..... 79
 - 5.14.3 **GET** 79
 - 5.14.3.1 *Example: get amount for volume split charge (Informative)* 79
 - 5.14.3.1.1 Request..... 79
 - 5.14.3.1.2 Response..... 80
 - 5.14.4 **PUT** 80
 - 5.14.5 **POST**..... 80
 - 5.14.6 **DELETE**..... 80
- 5.15 RESOURCE: ALL AMOUNT RESERVATION AND RELEASE TRANSACTIONS FOR AN END USER..... 80**
 - 5.15.1 **Request URI variables** 80
 - 5.15.2 **Response Codes** 81
 - 5.15.2.1 *HTTP Response Codes* 81
 - 5.15.2.2 *Exception fault codes*..... 81
 - 5.15.3 **GET** 81
 - 5.15.3.1 *Example: get all amount reservation transactions (Informative)* 81
 - 5.15.3.1.1 Request..... 81
 - 5.15.3.1.2 Response..... 81
 - 5.15.4 **PUT** 83
 - 5.15.5 **POST**..... 83
 - 5.15.5.1 *Example: create reserve amount (Informative)* 83
 - 5.15.5.1.1 Request..... 83
 - 5.15.5.1.2 Response..... 84
 - 5.15.6 **DELETE**..... 84
- 5.16 RESOURCE: INDIVIDUAL AMOUNT RESERVATION AND RELEASE TRANSACTION FOR AN END USER..... 85**
 - 5.16.1 **Request URI variables** 85
 - 5.16.2 **Response Codes** 85
 - 5.16.2.1 *HTTP Response Codes* 85
 - 5.16.2.2 *Exception fault codes*..... 85
 - 5.16.3 **GET** 85
 - 5.16.3.1 *Example: get amount reservation (Informative)* 85
 - 5.16.3.1.1 Request..... 85
 - 5.16.3.1.2 Response..... 86
 - 5.16.4 **PUT** 86

- 5.16.4.1 *Example 1: update (charge) amount for amount reservation (informative)* 86
 - 5.16.4.1.1 Request 86
 - 5.16.4.1.2 **Response** 87
- 5.16.4.2 *Example 2: update (release) amount for amount reservation (Informative)*..... 87
 - 5.16.4.2.1 Request 87
 - 5.16.4.2.2 Response..... 88
- 5.16.4.3 *Example 3: update (charge partial) amount for amount reservation (Informative)* 89
 - 5.16.4.3.1 Request 89
 - 5.16.4.3.2 Response..... 89
- 5.16.4.4 *Example 4: update (release) remaining amount reservation (Informative)*..... 90
 - 5.16.4.4.1 Request 90
 - 5.16.4.4.2 Response..... 90
- 5.16.4.5 *Example 5: update (reserve) additional amount for amount reservation (Informative)* 91
 - 5.16.4.5.1 Request 91
 - 5.16.4.5.2 Response..... 91
- 5.16.4.6 *Example 6 update (release) portion of amount reservation (Informative)*..... 92
 - 5.16.4.6.1 Request 92
 - 5.16.4.6.2 Response..... 93
- 5.16.5 **POST**..... 93
- 5.16.6 **DELETE**..... 93
- 5.17 RESOURCE: ALL VOLUME RESERVATION AND RELEASE TRANSACTIONS FOR AN END USER..... 93**
 - 5.17.1 **Request URI variables** 94
 - 5.17.2 **Response Codes** 94
 - 5.17.2.1 *HTTP Response Codes* 94
 - 5.17.2.2 *Exception fault codes*..... 94
 - 5.17.3 **GET** 94
 - 5.17.3.1 *Example: get all volume reservation transactions (Informative)*..... 94
 - 5.17.3.1.1 Request 94
 - 5.17.3.1.2 Response..... 94
 - 5.17.4 **PUT** 96
 - 5.17.5 **POST** 96
 - 5.17.5.1 **Example 1: create reserve volume (Informative)** 96
 - 5.17.5.1.1 Request 96
 - 5.17.5.1.2 **Response** 96
 - 5.17.5.2 **Example 2: create reserve volume with invalid endUserID (Informative)**..... 97
 - 5.17.5.2.1 Request 97
 - 5.17.5.2.2 **Response** 98
 - 5.17.6 **DELETE**..... 98
- 5.18 RESOURCE: INDIVIDUAL VOLUME RESERVATION AND RELEASE TRANSACTION FOR AN END USER..... 98**
 - 5.18.1 **Request URI variables** 98
 - 5.18.2 **Response Codes** 99
 - 5.18.2.1 *HTTP Response Codes* 99
 - 5.18.2.2 *Exception fault codes*..... 99
 - 5.18.3 **GET** 99
 - 5.18.3.1 *Example: get volume reservation (Informative)* 99
 - 5.18.3.1.1 Request 99
 - 5.18.3.1.2 Response..... 99
 - 5.18.4 **PUT** 100
 - 5.18.4.1 *Example: update charge volume (Informative)*..... 100
 - 5.18.4.1.1 **Request** 100
 - 5.18.4.1.2 **Response** 100
 - 5.18.5 **POST**..... 101
 - 5.18.6 **DELETE**..... 101
- 5.19 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME RESERVATION OR RELEASE TRANSACTION FOR AN END USER 101**
 - 5.19.1 **Request URI variables** 101
 - 5.19.2 **Response Codes** 102
 - 5.19.2.1 *HTTP Response Codes* 102
 - 5.19.2.2 *Exception fault codes*..... 102
 - 5.19.3 **GET** 102
 - 5.19.3.1 *Example: get amount for volume reservation (Informative)*..... 102
 - 5.19.3.1.1 Request 102

| | | |
|--------------------|---|------------|
| 5.19.3.1.2 | Response..... | 102 |
| 5.19.4 | PUT | 103 |
| 5.19.5 | POST..... | 103 |
| 5.19.6 | DELETE | 103 |
| 5.20 | RESOURCE: AMOUNT CONVERTED FROM GIVEN VOLUME | 103 |
| 5.20.1 | Request URI variables | 103 |
| 5.20.2 | Response Codes | 103 |
| 5.20.2.1 | <i>HTTP Response Codes</i> | 103 |
| 5.20.2.2 | <i>Exception fault codes</i> | 103 |
| 5.20.3 | GET | 104 |
| 5.20.3.1 | <i>Example: get amount converted from volume (Informative)</i> | 104 |
| 5.20.3.1.1 | Request..... | 104 |
| 5.20.3.1.2 | Response..... | 104 |
| 5.20.4 | PUT | 104 |
| 5.20.5 | POST..... | 105 |
| 5.20.6 | DELETE | 105 |
| APPENDIX A. | CHANGE HISTORY (INFORMATIVE) | 106 |
| A.1 | APPROVED VERSION HISTORY | 106 |
| A.2 | DRAFT VERSION 1.0 HISTORY | 106 |
| APPENDIX B. | STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) | 108 |
| B.1 | SCR FOR PARLAYREST.PAYMENT SERVER | 108 |
| B.1.1 | SCR for ParlayREST.Payment.Transactions Server | 108 |
| B.1.2 | SCR for ParlayREST.Payment.AmountCharge.Transactions Server | 108 |
| B.1.3 | SCR for ParlayREST.Payment.SplitAmount.Transactions Server | 108 |
| B.1.4 | SCR for ParlayREST.Payment.Individual.AmountCharge Server | 109 |
| B.1.5 | SCR for ParlayREST.Payment.Individual.SplitAmount Server | 109 |
| B.1.6 | SCR for ParlayREST.Payment.VolumeCharge.Transactions Server..... | 109 |
| B.1.7 | SCR for ParlayREST.Payment.Split.Volume.Transactions Server..... | 110 |
| B.1.8 | SCR for ParlayREST.Payment.Individual.VolumeCharge Server..... | 110 |
| B.1.9 | SCR for ParlayREST.Payment.Individual.Split.VolumeCharge Server | 110 |
| B.1.10 | SCR for ParlayREST.Payment.Individual.Amount.VolumeCharge Server..... | 111 |
| B.1.11 | SCR for ParlayREST.Payment.Individual.Amount.Split.VolumeCharge Server | 111 |
| B.1.12 | SCR for ParlayREST.Payment.Amount.Reserve.Transactions Server | 111 |
| B.1.13 | SCR for ParlayREST.Payment.Amount.Individual.Reserve.Transactions Server | 111 |
| B.1.14 | SCR for ParlayREST.Payment.Volume.Reserve.Transactions Server | 112 |
| B.1.15 | SCR for ParlayREST.Payment.Volume.Individual.Reserve.Transactions Server | 112 |
| B.1.16 | SCR for ParlayREST.Payment.Individual.Amount.Volume Server | 112 |
| B.1.17 | SCR for ParlayREST.Payment.Amount.Converted Server..... | 112 |
| APPENDIX C. | APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR SELECTED OPERATIONS | 114 |
| C.1 | CHARGE AN AMOUNT | 114 |
| C.1.1 | Example (Informative)..... | 115 |
| C.1.1.1 | Request..... | 115 |
| C.1.1.2 | Response | 116 |
| C.2 | REFUND AN AMOUNT | 116 |
| C.2.1 | Example (Informative)..... | 118 |
| C.2.1.1 | Request..... | 118 |
| C.2.1.2 | Response | 118 |
| C.3 | RESERVE AN AMOUNT | 119 |
| C.3.1 | Example (Informative)..... | 120 |
| C.3.1.1 | Request..... | 120 |
| C.3.1.2 | Response | 121 |
| C.4 | RESERVE AN ADDITIONAL AMOUNT | 121 |
| C.4.1 | Example (Informative)..... | 123 |
| C.4.1.1 | Request..... | 123 |
| C.4.1.2 | Response | 123 |
| C.5 | CHARGE TO A RESERVATION | 124 |

C.5.1 Example (Informative)..... 125

 C.5.1.1 Request..... 125

 C.5.1.2 Response..... 126

C.6 RELEASE A RESERVATION127

 C.6.1 Example (Informative)..... 128

 C.6.1.1 Request..... 128

 C.6.1.2 Response..... 128

APPENDIX D. JSON EXAMPLES (INFORMATIVE)130

 D.1.1.1 Get all transactions (section 5.4.3.1.1) 130

 D.1.1.2 Request with invalid endUserId (section 5.4.3.2)..... 134

 D.1.1.3 Get all amount transactions (section 5.5.3.1) 134

 D.1.1.4 Create charge amount (section 5.5.5.1) 136

 D.1.1.5 Create refund amount (section 5.5.5.2)..... 136

 D.1.1.6 Client retries POST with same clientCorrelator (section 5.5.5.3) 137

 D.1.1.7 Get all amount split transactions (section 5.6.3.1)..... 138

 D.1.1.8 Create split charge amount (section 5.6.5.1) 140

 D.1.1.9 Get amount charge (Section 5.7.3.1)..... 141

 D.1.1.10 Update amount transaction (section 5.7.4.1)..... 141

 D.1.1.11 Get amount split charge (section 5.8.3.1) 142

 D.1.1.12 Update split charge amount transaction (section 5.8.4.1)..... 143

 D.1.1.13 Get all volume charge and refund transactions (section 5.9.3.1) 144

 D.1.1.14 Create charge volume, returning a representation of created resource (section 5.9.5.1) 145

 D.1.1.15 Create charge volume, returning the location of created resource (section 5.9.5.2)..... 146

 D.1.1.16 Create refund volume (section 5.9.5.2.3)..... 147

 D.1.1.17 Get all volume split charge transactions (section 5.10.3.1)..... 148

 D.1.1.18 Create volume split charge (section 5.10.5.1) 149

 D.1.1.19 Get volume charge (section 5.11.3.1)..... 150

 D.1.1.20 Update volume charge transaction (section 5.11.4.1) 151

 D.1.1.21 Get volume split charge (section 5-12-3-1) 151

 D.1.1.22 Update volume split charge transaction (section 5.12.4.1) 152

 D.1.1.23 Get amount for volume charge (section 5.13.3.1)..... 154

 D.1.1.24 Get amount for volume split charge (section 5.14.3.1)..... 154

 D.1.1.25 Get all amount reservation transactions (section 5.15.3.1) 155

 D.1.1.26 Create reserve amount (section 5.15.5.1)..... 156

 D.1.1.27 Get amount reservation (section 5.16.3.1)..... 157

 D.1.1.28 Update (charge) amount for amount reservation (section 5.16.4.1)..... 158

 D.1.1.29 Update (release) amount for amount reservation..... 159

 D.1.1.30 Update (charge partial) amount for amount reservation (section 5.16.4.3) 160

 D.1.1.31 Update (release) remaining amount reservation (section 5.16.4.4) 161

 D.1.1.32 Update (reserve) additional amount for amount reservation (section 5.16.4.5)..... 162

 D.1.1.33 Update (release) portion of amount reservation (section 5.16.4.6)..... 163

 D.1.1.34 Get all volume reservation transactions (section 5.17.3.1) 164

 D.1.1.35 **Create** reserve volume (section 5.17.5.1)..... 165

 D.1.1.36 **Create** reserve volume with invalid endUserID (section 5.17.5.2)..... 166

 D.1.1.37 Get volume reservation (section 5.18.3.1)..... 167

 D.1.1.38 Update charge volume (section 5.18.4.1) 168

 D.1.1.39 Get amount for volume reservation (section 5.19.3.1)..... 169

 D.1.1.40 Get amount converted from volume (section 5.20.3.1) 169

Figures

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

Figure 2 Amount charge and refund transaction..... 33

Figure 3 Amount split charge transaction 34

Figure 4 Volume charge and refund transaction 35

Figure 5 Volume split charge transaction..... 36

Figure 6 Amount reservation and release transaction..... 37

Figure 7 Volume reservation and release transaction39
Figure 8 Amount converted from volume transaction40

1. Scope

This specification defines an HTTP protocol binding for an abstract API using the REST architectural style , based on existing OMA enabler namely the Payment part, as defined in [3GPP 29.199-6].

2. References

2.1 Normative References

- [3GPP 29.199-5] 3GPP Technical Specification, “Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (Release 8)”, URL:<http://www.3gpp.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>
- [REST_TS_Common] “RESTful bindings for Parlay X Web Services – Common”, Open Mobile Alliance™, OMA-TS-ParlayREST_Common-V1_0, URL:<http://www.openmobilealliance.org/>
- [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

- [OMADICT] “Dictionary for OMA Specifications”, Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, 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 [OMA-DICT].

[N/A]

[N/A]

3.3 Abbreviations

| | |
|---------------|-----------------------------------|
| ACR | Anonymous Customer Reference |
| API | Application Programming Interface |
| HTTP | HyperText Transfer Protocol |
| ID | Identifier |
| JSON | JavaScript Object Notation |
| MSISDN | Mobile Subscriber ISDN Number |
| OMA | Open Mobile Alliance |
| PX | Parlay X |
| 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 a charge amount or volume for an end user's account
- Adding/subtracting a charge amount or volume to/from an existing reservation
- Charging to a previously made reservation
- Releasing funds left in a previously made reservation

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].

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). In addition, for each supported resource/verb combination, the table lists the Parlay X equivalent operation, where applicable. 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).

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.

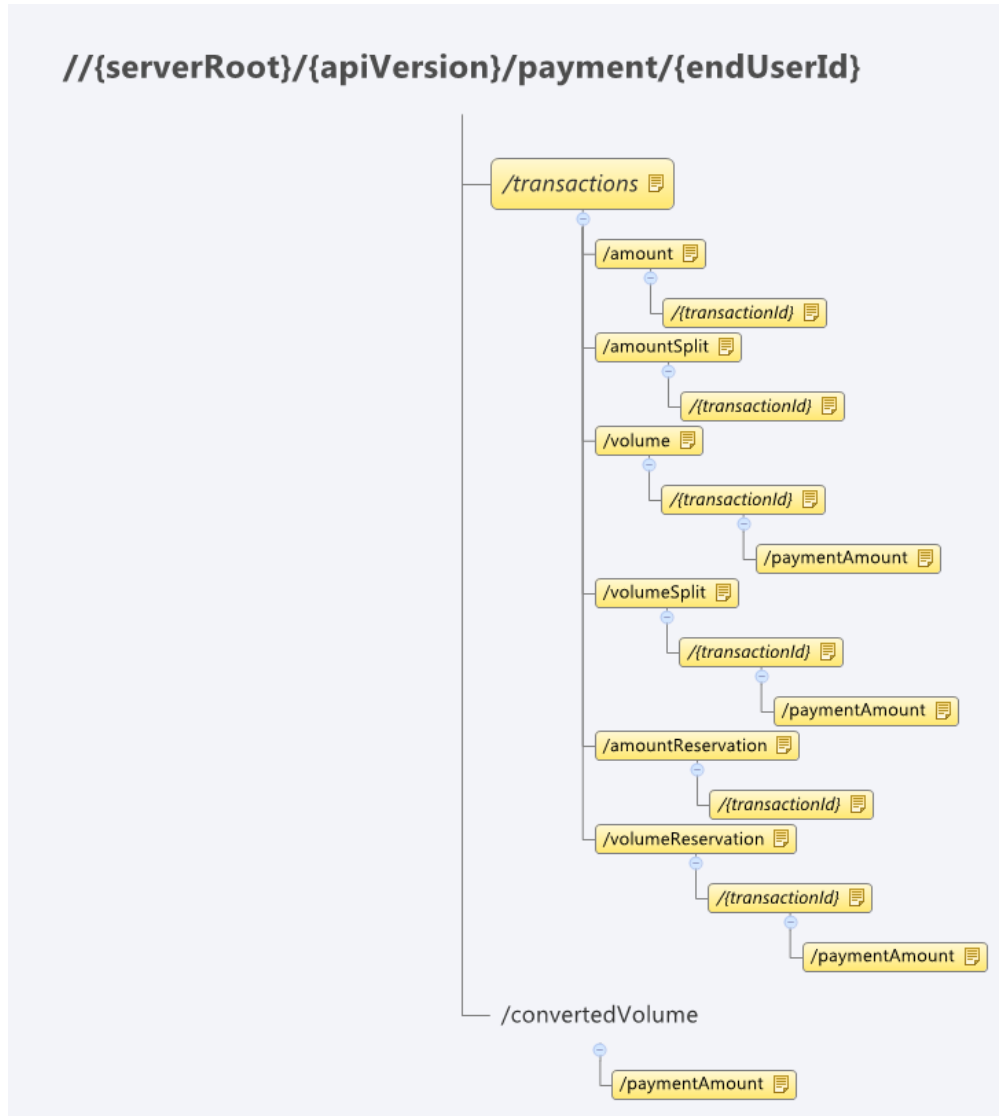


Figure 1 Resource structure defined by this specification

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. The “PX” row indicates the Parlay X SOAP equivalent operation.

Purpose: All payment transactions

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|-----------------------------------|--|------------------------|---|------|-----|--------|
| | | | GET | POST | PUT | 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 |
| | | | No PX equivalent | | | |

Purpose: Amount charge, split charge, refund transactions

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|---|--|------------------------|---|---|-----|--------|
| | | | GET | POST | PUT | DELETE |
| All amount charge and refund transactions for an end user | /{endUserId}/transactions/amount | PaymentTransactionList | return all amount transactions details and state for a given end user | create new transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ChargeAmount, RefundAmount | | |
| All amount split charge transactions for an end user | /{endUserId}/transactions/amountSplit | PaymentTransactionList | return all amount split transactions details and state for a given end user | create new transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ChargeSplitAmount | | |

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|--|--|------------------------|--|------|---|--------|
| | | | GET | POST | PUT | DELETE |
| Individual amount charge or refund transaction for an end user | /{endUserId}/transactions/amount/{transactionId} | AmountTransaction | return amount transaction details (charge, refund) | no | update transaction information and state - depends on the state transaction is currently in | no |
| | | | No PX equivalent | | No PX equivalent | |
| Individual amount split charge transaction for an end user | /{endUserId}/transactions/amountSplit/{transactionId} | AmountSplitTransaction | return amount split transaction details (charge) | no | update transaction information and state - depends on the state transaction is currently in | no |
| | | | No PX equivalent | | No PX equivalent | |

Purpose: Volume charge, split charge, refund transactions

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|---|--|------------------------|---|---|-----|--------|
| | | | GET | POST | PUT | DELETE |
| All volume charge and refund transactions for an end user | /{endUserId}/transactions/volume | PaymentTransactionList | return all volume transactions details and state for a given end user | create new transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ChargeVolume, RefundVolume | | |

| Resource | URL Base URL: http://{serverRoot}/{apiVersion}/payment | Data Structures | HTTP verbs | | | |
|---|--|------------------------|---|---|---|--------|
| | | | GET | POST | PUT | DELETE |
| All volume split charge transactions for an end user | /{endUserId}/transactions/volumeSplit | PaymentTransactionList | return all volume split transactions details and state for a given end user | create new transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ChargeSplitVolume | | |
| Individual volume charge or refund transaction for an end user | /{endUserId}/transactions/volume/{transactionId} | VolumeTransaction | return volume transaction details (charge, refund) | no | update transaction information and state - depends on the state transaction is currently in | no |
| | | | No PX equivalent | | No PX equivalent | |
| Individual volume split charge transaction for an end user | /{endUserId}/transactions/volumeSplit/{transactionId} | VolumeSplitTransaction | return volume split transaction details (charge) | no | update transaction information and state - depends on the state transaction is currently in | no |
| | | | No PX equivalent | | No PX equivalent | |
| Individual amount for volume charge or refund transaction for an end user | /{endUserId}/transactions/volume/{transactionId}/paymentAmount | PaymentAmount | return transaction payment information based on provided volume | no | no | no |
| | | | No PX equivalent | | | |

| Resource | URL Base URL: http://{serverRoot}/{apiVersion}/payment | Data Structures | HTTP verbs | | | |
|---|---|-----------------|---|------|-----|--------|
| | | | GET | POST | PUT | DELETE |
| 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 |
| | | | No PX equivalent | | | |

Purpose: Amount reservation and release transactions

| Resource | URL Base URL: http://{serverRoot}/{apiVersion}/payment | Data Structures | HTTP verbs | | | |
|---|--|------------------------|---|---|-----|--------|
| | | | GET | POST | PUT | DELETE |
| All amount reservation and release transactions for an end user | /{endUserId}/transactions/amountReservation | PaymentTransactionList | return all amount reservation transactions details and state for a given end user | create new reservation transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ReserveAmount | | |

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|---|--|------------------------------|--|------|--|--------|
| | | | GET | POST | PUT | DELETE |
| Individual amount reservation and release transaction for an end user | /{endUserId}/transactions/amountReservations/{transactionId} | AmountReservationTransaction | return amount reservation transaction details (amount, charge) | no | change transaction state to "charged", increase reservation amount, release reserved amount (state = "released") | no |
| | | | No PX equivalent | | PX: ReserveAdditionalAmount, ChargeReservation, ReleaseReservation | |

Purpose: Volume reservation and release transactions

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|--|---|------------------------------|---|--|---|--------|
| | | | GET | POST | PUT | DELETE |
| All volume reservation and release transactions for an end user | /{endUserId}/transactions/volumeReservation | PaymentTransactionList | return transaction details and state for a given end user | create new volume reservation transaction for a given end user | no | no |
| | | | No PX equivalent | PX: ReserveVolume | | |
| Individual volume reservation and release transaction for an end user | /{endUserId}/transactions/volumeReservation/{transactionId} | VolumeReservationTransaction | return volume reservation transaction details (volume charge) | no | change transaction state to "charged", increase reservation volume or release reserved volume amount (state = "released") | no |
| | | | No PX equivalent | | PX: ReserveAdditionalVolume, ChargeReservation, ReleaseReservation | |
| Individual amount for volume reservation and release transaction for an end user | /{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount | PaymentAmount | return transaction payment information (calculates charge amount based on volume) | no | no | no |
| | | | No PX equivalent | | | |

Purpose: Amount converted from volume transaction

| Resource | URL Base URL: http://{serverRoot}/{api Version}/payment | Data Structures | HTTP verbs | | | |
|------------------------------------|---|-----------------|--|------|-----|--------|
| | | | GET | POST | PUT | 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 |
| | | | PX: GetAmount | | | |

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 [REST_TS_Common]. The use of the names 'xsd' and 'common' is not semantically significant.

5.2.1 Type: PaymentTransactionList

| 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 |
| 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

| 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 |
| transactionStatus | TransactionStatus | 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. |
| 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 |
| 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

| 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 |
| transactionStatus | TransactionStatus | 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. |
| 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 |
| 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

| 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 |
| transactionStatus | TransactionStatus | 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. |
| 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 |
| 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

| 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 |
| transactionStatus | TransactionStatus | 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. |
| 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</p> |

| | | | |
|-------------|---------------------------|-----|--|
| | | | 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 |
| 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

| 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 |
| transactionStatus | TransactionStatus | No | E.g. charged, refunded, etc |
| 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). |
| 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. |

| | | | |
|------------------|---------------------------|-----|---|
| 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 |
| 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

| Element | Type | Optional | Description |
|-------------------|-------------------|----------|---|
| endUserId | xsd:anyURI | No | The end user's account to be charged |
| paymentVolume | PaymentVolume | No | Information on the amount charge to be made |
| transactionStatus | TransactionStatus | No | E.g. charged, refunded, etc |
| referenceSequence | xsd:integer | 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</p> |

| | | | |
|------------------|---------------------------|-----|---|
| | | | 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. |
| 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 |
| 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

| 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

| 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

| 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

| 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

| 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: TransactionStatus

| Enumeration | Description |
|-------------|--|
| Charging | The amount or volume is being charged |
| Charged | The amount or volume has been charged |
| Refunding | The amount or volume is being refunded |
| Refunded | The amount or volume has been refunded |
| Reserving | The amount or volume is being reserved |
| Reserved | The amount or volume has been reserved |
| Denied | The reservation is denied |
| Cancelled | The reservation is cancelled |
| Refused | The reservation is refused |
| Released | The reservation is released |

Note: The following transaction states are supported for amount or volume charge or refund transactions:

- “charging” → “charged” (success)
- “charging” → “denied” (insufficient balance, security, etc)
- “charging” → “refused” (rejected by end user – confirmation is not received)
- “refunding” → “refunded” (success)
- “refunding” → “denied” (insufficient balance, security, etc)
- “refunding” → “refused” (rejected by end user - confirmation is not received)

Note: The following transaction states are supported for amount or volume reservation or release transactions:

- “releasing” → “released” (success)
- “releasing” (remaining) → “released” (remaining)
- “reserving” → “reserved” (success)
- “reserving” → “denied” (insufficient balance, security, etc)
- “reserving” → “refused” (rejected by end user – confirmation is not received)
- “reserving” (additional amount) → “reserved” (success)
- “charging” (additional amount) → “charged” (success)
- “charging” (partial amount) → “charged” (success)

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`**

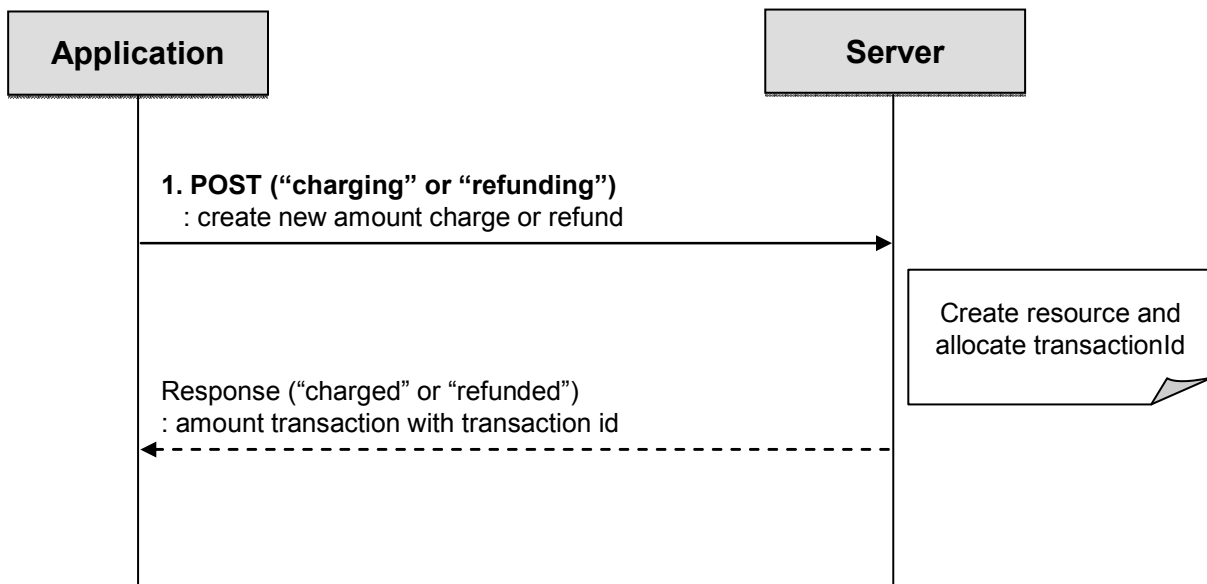


Figure 2 Amount charge and refund transaction

Outline of flow:

- 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

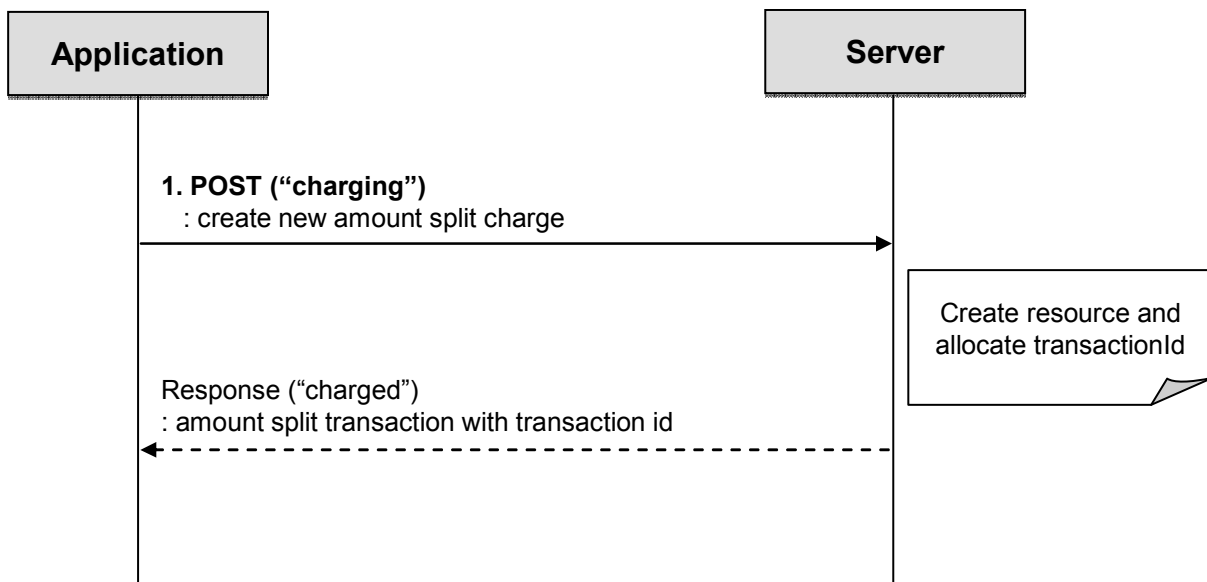


Figure 3 Amount split charge transaction

Outline of flow:

- 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

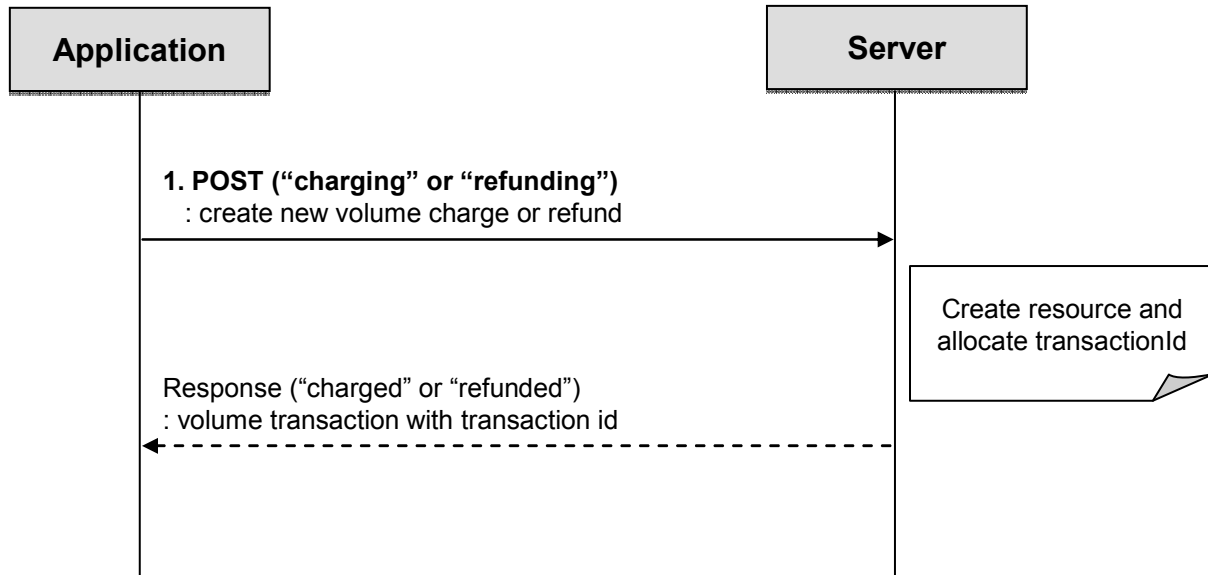


Figure 4 Volume charge and refund transaction

Outline of flow:

- 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`**

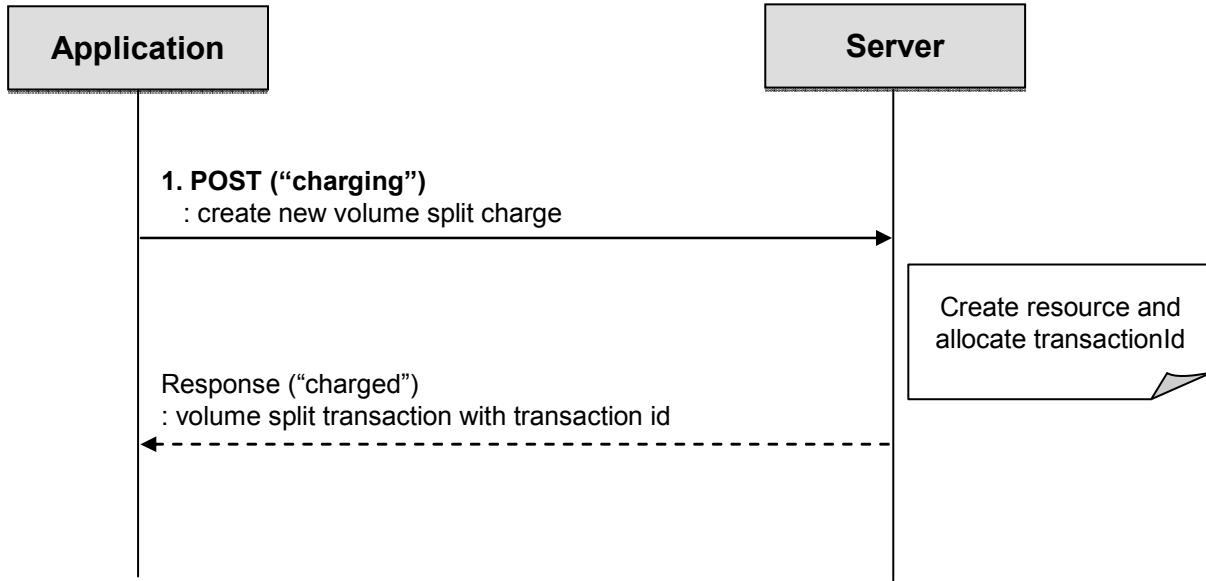


Figure 5 Volume split charge transaction

Outline of flow:

- 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 and release transaction

This figure below shows a scenario to create an amount reservation, charge and release 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 create an amount reservation transaction for an end user, use resource
`http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}`

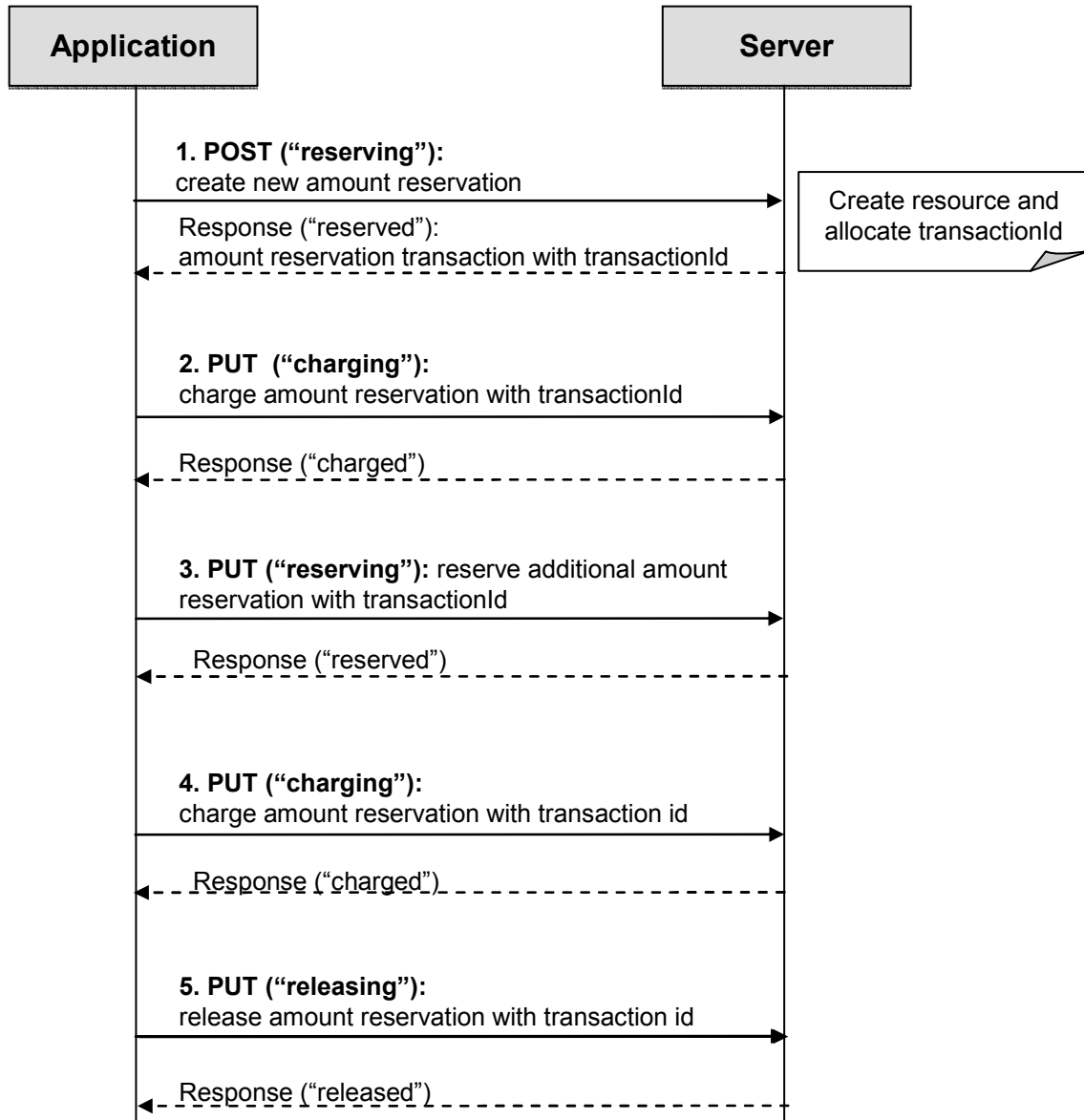


Figure 6 Amount reservation and release transaction

Outline of flow:

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

5.3.6 Volume reservation and release transaction

This figure below shows a scenario to create a volume reservation and release 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 create resource
`http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}`

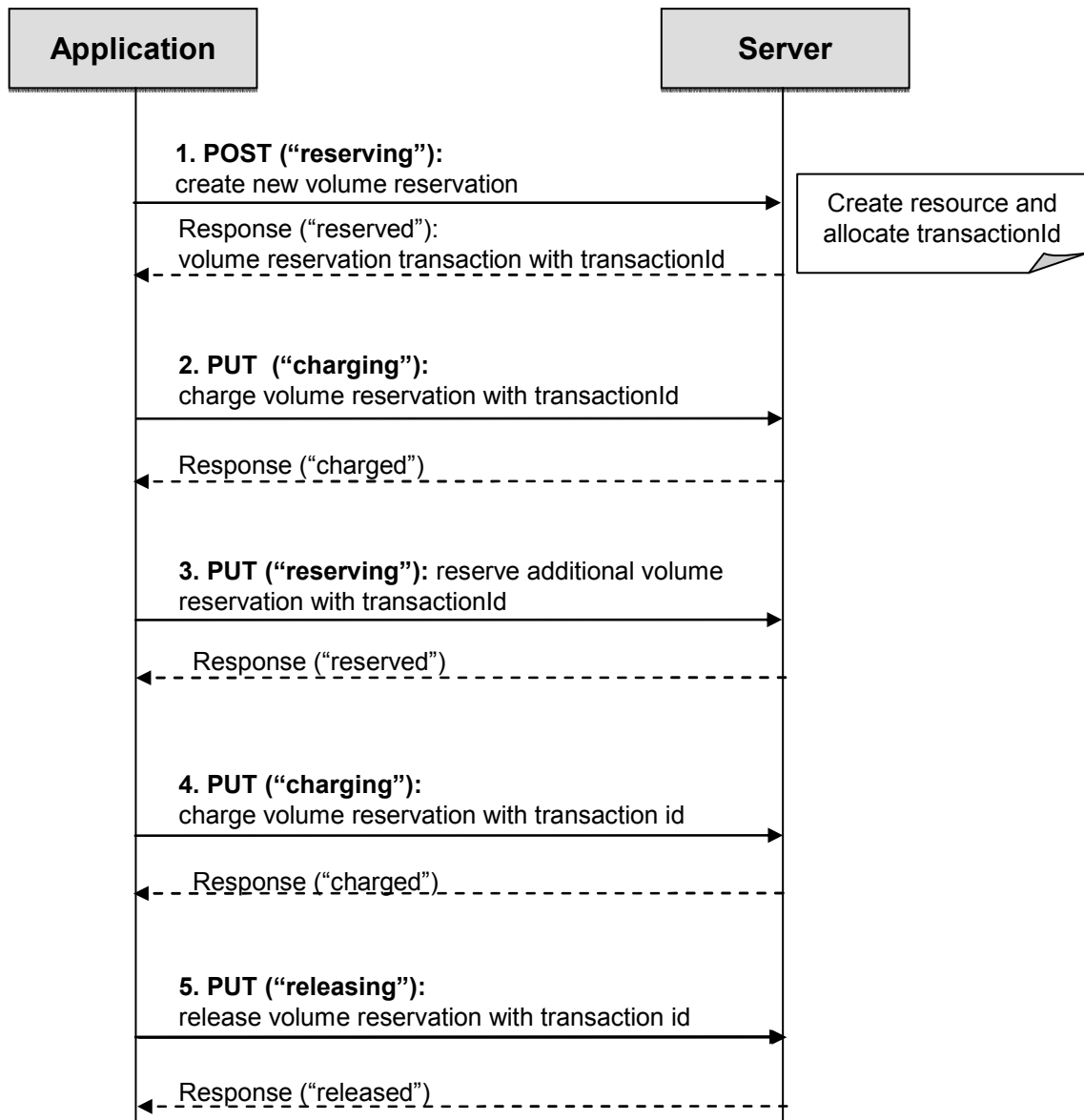


Figure 7 Volume reservation and release transaction

Outline of flow:

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

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`

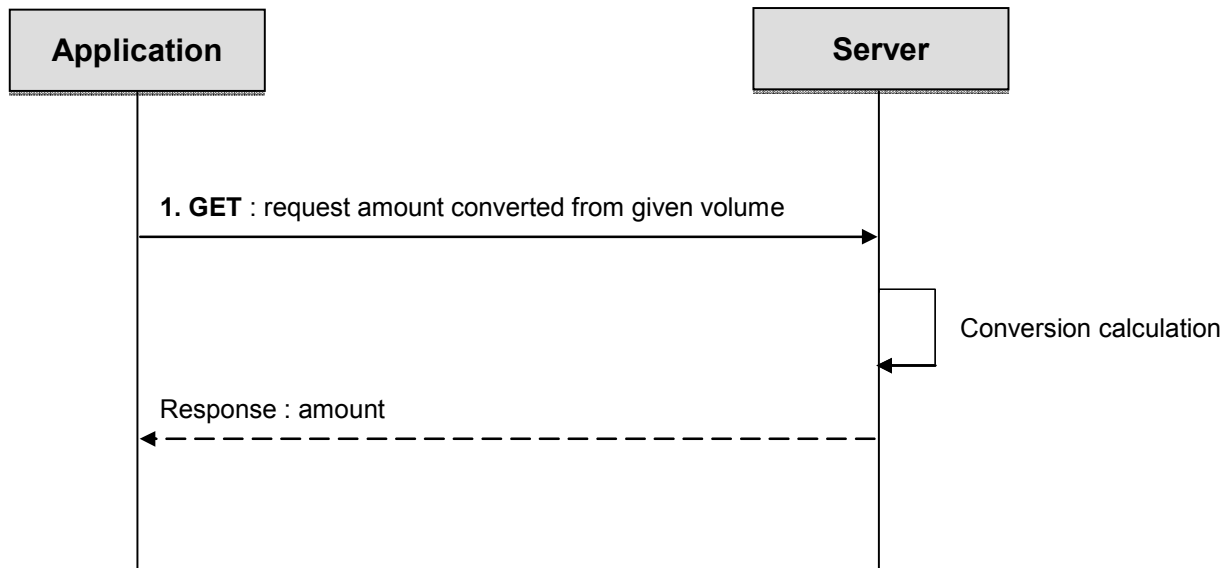


Figure 8 Amount converted from volume transaction

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`**

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.4.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

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.

Note: No ParlayX SOAP equivalent.

5.4.3.1 Example 1: get all transactions

(Informative)

5.4.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.4.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
```

```

    <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT REFUND TRANSACTION -->
<amountTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54322</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
<amountSplitTransaction>
  <endUserShare>
    <endUserId>tel:+16309700001</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54323</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
</amountSplitTransaction>

```

```
<!-- COMPLETED VOLUME CHARGE TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charge" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</volumeTransaction>
<!-- COMPLETED VOLUME REFUND TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Refunded" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54324</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</volumeTransaction>
<!-- COMPLETED VOLUME SPLIT CHARGE TRANSACTION -->
<volumeSplitTransaction>
  <endUserShare>
    <endUserId>tel:+16309700001</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
```

```

<clientCorrelator>54325</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
</volumeSplitTransaction>
<!--
VOLUME RESERVATION TRANSACTION IN "RESERVED" STATE
-->
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>

  <paymentVolume>
    <billingText>Test volume reservation transaction in "Reserved" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>
<!-- VOLUME RESERVATION TRANSACTION IN "CHARGED" STATE -->
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66667</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>
<!--
COMPLETED AMOUNT RESERVATION TRANSACTION
-->
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>

```

```

</chargingInformation>
<totalAmountCharged>25</totalAmountCharged>
<amountReserved>0</amountReserved>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

</amountReservationTransaction>
<!-- AMOUNT RESERVATION TRANSACTION IN "RESERVED" STATE -->
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <!-- last additional reserve applied -->
  <paymentAmount>
    <chargingInformation>
      <description>Test reserve amount</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>

</amountReservationTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions</resourceURL>
</payment:paymentTransactionList>

```

5.4.3.2 Example 2: request with invalid endUserId

(Informative)

5.4.3.2.1 Request

```

GET ../{apiVersion}/payment/{endUserId}/transactions HTTP/1.1
Accept: application/xml
Host: example.com:80

```

5.4.3.2.2 Response

```

HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: 12345
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://{serverRoot}/apiVersion}/payment/{endUserId}/transactions"/>
<serviceException>
  <messageId>SVC0004</messageId>
  <text>Invalid input value</text>
  <variables>The tel:+016309700000 is not valid</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`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.5.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.5.3 GET

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

Note: No ParlayX SOAP equivalent.

5.5.3.1 Example: get all amount transactions

(Informative)

5.5.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.5.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>54321</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
  </amountTransaction>
  <!-- COMPLETED AMOUNT CHARGE TRANSACTION - ACRed end user id-->
  <amountTransaction>
    <endUserId>tel:1234567890</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
```

```

    <amount>10</amount>
    <code>TEST012345</code>
  </chargingInformation>
</totalAmountCharged>10</totalAmountCharged>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT REFUND TRANSACTION -->
<amountTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54330</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>

  </amountTransaction>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/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.

Note: ParlayX SOAP equivalents are ChargeAmount and RefundAmount.

5.5.5.1 Example 1: create charge amount

(Informative)

5.5.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/xml

```



```

Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <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: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</payment:amountTransaction>

```

5.5.5.2 Example 2: create refund amount**(Informative)****5.5.5.2.1 Request**

```

POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

```

5.5.5.2.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</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 [TS_Common] section 5.6.1 for other possible cases.

5.5.5.3.1 Request

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <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: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
```

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

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

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.6.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.6.3 GET

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

Note: No ParlayX SOAP equivalent.

5.6.3.1 Example: get all amount split transactions**(Informative)****5.6.3.1.1 Request**

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amountSplit HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.6.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>30</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700001</endUserId>
      <percent>70</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
        <amount>20</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>20</totalAmountCharged>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55552</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
  </amountSplitTransaction>
  <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
  <amountSplitTransaction>
    <endUserShare>
      <endUserId>tel:+16309700001</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700002</endUserId>
      <percent>80</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
```

```

<description>Test amount transaction in "Charged" state</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST-012346</code>
</chargingInformation>
<totalAmountCharged>10</totalAmountCharged>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<referenceCode>REF-12346</referenceCode>
<clientCorrelator>55553</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
</amountSplitTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/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.

Note: ParlayX SOAP equivalent is ChargeSplitAmount.

5.6.5.1 Example: create split charge amount

(Informative)

5.6.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/amountSplit HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <percent>30</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700001</endUserId>
    <percent>70</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
    </chargingInformation>
  </paymentAmount>
</payment:amountSplitTransaction>

```

```

    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<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: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}

```

```

<?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:+16309700001</endUserId>
    <percent>30</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>70</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54431</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</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}`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.7.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.7.3 GET

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

Note: No ParlayX SOAP equivalent.

5.7.3.1 Example: get amount charge (Informative)

Note: this example also illustrates how to indicate in the request the expected response body format.

5.7.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com:80
```


5.7.3.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>54321</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
  </payment:amountTransaction>

```

5.7.4 PUT

This operation is used to update an individual end user's charge amount or refund amount transaction.

Note: No ParlayX SOAP equivalent.

5.7.4.1 Example: update amount transaction

(Informative)

5.7.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
    </chargingInformation>
  </paymentAmount>
</payment:amountTransaction>

```

```

    <code>TEST-012345</code>
  </chargingInformation>
</totalAmountRefunded>10</totalAmountRefunded>
</paymentAmount>
<transactionStatus>Refunded</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

```

5.7.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</payment:amountTransaction>

```

5.7.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT' 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, PUT' 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}`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.8.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.8.3 GET

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

Note: No ParlayX SOAP equivalent.

5.8.3.1 Example: get amount split charge

(Informative)

5.8.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}
HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.8.3.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>30</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700002</endUserId>
      <percent>70</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>54321</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
  </payment:amountSplitTransaction>

```

5.8.4 PUT

This operation is used to update an individual end user's split charge amount transaction.

Note: No ParlayX SOAP equivalent.

5.8.4.1 Example: update split charge amount transaction (Informative)

5.8.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <percent>30</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>70</percent>
  </endUserShare>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountSplitTransaction>

```

5.8.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>30</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700002</endUserId>
      <percent>70</percent>
    </endUserShare>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount transaction in "Charged" state</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>54321</clientCorrelator>
  </payment:amountSplitTransaction>

```

```
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
</payment:amountSplitTransaction>
```

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, PUT’ 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, PUT’ 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**

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.9.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.9.3 GET

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

Note: No ParlayX SOAP equivalent.

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

5.9.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.9.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction in "Charged" state</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55551</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}
  </resourceURL>

  </volumeTransaction>
  <!-- COMPLETED VOLUME REFUND TRANSACTION -->
  <volumeTransaction>
    <endUserId>tel:+16309700001</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction in "Refunded" state</billingText>
      <volume>10</volume>
      <ratingParameter>
```

```

    <name>unit</name>
    <value>minutes</value>
  </ratingParameter>
</totalVolumeRefunded>10</totalVolumeRefunded>
</paymentVolume>
<transactionStatus>Refunded</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55552</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}
</resourceURL>
</volumeTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/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.

Note: ParlayX SOAP equivalents are ChargeVolume and RefundVolume.

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

5.9.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com:80
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <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://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</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 ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com:80
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <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://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}
Content-Length: 254
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://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</common:resourceReference>
```

5.9.5.2.3 Example 3: create refund volume

(Informative)

5.9.5.2.4 Request

```
POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com:80
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Refunded" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55552</clientCorrelator>
</payment:volumeTransaction>
```

5.9.5.2.5 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}
```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME REFUND TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentVolume>
    <billingText>Test volume transaction in "Refunded" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55552</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</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>

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.10.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.10.3 GET

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

Note: No ParlayX SOAP equivalent.

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

5.10.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit
HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.10.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700002</endUserId>
      <percent>80</percent>
    </endUserShare>
    <paymentVolume>
      <billingText>Test volume transaction in "Charged" state</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
```

```

<transactionStatus>Charged</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55553</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>

</volumeSplitTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit</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.

Note: ParlayX SOAP equivalent is ChargeSplitVolume.

5.10.5.1 Example: create volume split charge

(Informative)

5.10.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
</transactionStatus>Charged</transactionStatus>

```

```

<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: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME SPLIT CHARGE TRANSACTION -->
<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserShare>
    <endUserId>tel:+16309700001</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55553</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</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}

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.11.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.11.3 GET

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

Note: No ParlayX SOAP equivalent.

5.11.3.1 Example: get volume charge

(Informative)

5.11.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.11.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</payment:volumeTransaction>

```

5.11.4 PUT

This operation is used to update an individual end user's volume amount or refund amount transaction.

Note: No ParlayX SOAP equivalent.

5.11.4.1 Example: update volume charge transaction

(Informative)

5.11.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55555</clientCorrelator>

```



```
</payment:volumeTransaction>
```

5.11.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentVolume>
      <billingText>Test volume transaction in "Charged" state</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
  </payment:volumeTransaction>
```

5.11.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, PUT’ 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, PUT’ 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}

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.12.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.12.3 GET

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

Note: No ParlayX SOAP equivalent.

5.12.3.1 Example: get volume split charge

(Informative)

5.12.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.12.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>20</percent>
    </endUserShare>
    <endUserShare>
      <endUserId>tel:+16309700002</endUserId>
      <percent>80</percent>
    </endUserShare>

    <paymentVolume>
      <billingText>Test volume transaction in "Charged" state</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionStatus>Charged</transactionStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55556</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
  </payment:volumeSplitTransaction>

```

5.12.4 PUT

This operation is used to update an individual end user's volume split charge transaction.

Note: No ParlayX SOAP equivalent.

5.12.4.1 Example: update volume split charge transaction (Informative)

5.12.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
      <percent>20</percent>
    </endUserShare>

```

```

<endUserShare>
  <endUserId>tel:+16309700002</endUserId>
  <percent>80</percent>
</endUserShare>
<paymentVolume>
  <billingText>Test volume transaction in "Charged" state</billingText>
  <volume>10</volume>
  <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>10</totalVolumeCharged>
</paymentVolume>
<transactionStatus>Charged</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55556</clientCorrelator>
</payment:volumeSplitTransaction>

```

5.12.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <percent>20</percent>
  </endUserShare>
  <endUserShare>
    <endUserId>tel:+16309700002</endUserId>
    <percent>80</percent>
  </endUserShare>
  <paymentVolume>
    <billingText>Test volume transaction in "Charged" state</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
</payment:volumeSplitTransaction>

```

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, PUT' 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, PUT' 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`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.13.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.13.3 GET

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

Note: No ParlayX SOAP equivalent.

5.13.3.1 Example: get amount for volume charge (Informative)

5.13.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/volume/{transactionId}/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.13.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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={transactionId} of endUserId={endUserId}</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`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.14.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.14.3 GET

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

Note: No ParlayX SOAP equivalent.

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

5.14.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.14.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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={transactionId}</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 and release transactions for an end user

The resource used is:

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

This resource is used to provide access to all the completed and pending amount reservation and release 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:80/ParlayREST |
| 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 [REST_TS_Common].

5.15.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.15.3 GET

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

Note: No ParlayX SOAP equivalent.

5.15.3.1 Example: get all amount reservation transactions (Informative)

5.15.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amountReservation HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.15.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>

    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction in "Charged" state</description>
        <currency>USD</currency>
```

```

    <amount>15</amount>
    <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>25</totalAmountCharged>
  <amountReserved>0</amountReserved>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</amountReservationTransaction>
<!-- AMOUNT RESERVATION TRANSACTION IN "RESERVED" STATE -->
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test reserve amount</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</amountReservationTransaction>
<!--
  AMOUNT RESERVATION TRANSACTION IN "RESERVED" STATE AFTER ADDITIONAL
  RESERVATION AMOUNT APPLIED
-->
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test reserve amount</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</amountReservationTransaction>
<!--
  AMOUNT RESERVATION TRANSACTION IN "RESERVED" STATE AFTER ADDITIONAL
  RESERVATION AMOUNT IS APPLIED SECOND TIME
-->

```

```

-->
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test reserve amount</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>25</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</amountReservationTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/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.

Note: ParlayX SOAP equivalent is ReserveAmount.

5.15.5.1 Example: create reserve amount

(Informative)

5.15.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/amountReservation HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>

```

```

<description>Test amount reservation transaction</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST-012345</code>
</chargingInformation>
</paymentAmount>
<transactionStatus>Reserved</transactionStatus>
<referenceSequence>1</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
</payment:amountReservationTransaction>

```

5.15.5.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in "Reserved" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</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 and release transaction for an end user

The resource used is:

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

This resource is used to provide access to an individual amount reservation or release 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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.16.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.16.3 GET

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

Note: No ParlayX SOAP equivalent.

5.16.3.1 Example: get amount reservation

(Informative)

5.16.3.1.1 Request

```
GET ../{apiVersion}/{endUserId}/transactions/amountReservation/{transactionId}
HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.16.3.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>0</totalAmountCharged>
      <amountReserved>10</amountReserved>
    </paymentAmount>
    <transactionStatus>Reserved</transactionStatus>
    <referenceSequence>1</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
  </payment:amountReservationTransaction>

```

5.16.4 PUT

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

Note: ParlayX SOAP equivalent are ReserveAdditionalAmount, ChargeReservation, ReleaseReservation.

5.16.4.1 Example 1: update (charge) amount for amount reservation (informative)

5.16.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>

```

```

<amount>10</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountCharged>10</totalAmountCharged>
<amountReserved>0</amountReserved>
</paymentAmount>
<transactionStatus>Charged</transactionStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>55555</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
      <amountReserved>0</amountReserved>
    </paymentAmount>
    <transactionStatus>Charged</transactionStatus>
    <referenceSequence>2</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
  </payment:amountReservationTransaction>

```

5.16.4.2 Example 2: update (release) amount for amount reservation (Informative)

5.16.4.2.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Released</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12346</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.2.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionStatus>Released</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12346</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```


5.16.4.3 Example 3: update (charge partial) amount for amount reservation (Informative)

5.16.4.3.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12347</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.3.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>5</amountReserved>
  </paymentAmount>

```

```

<transactionStatus>Charged</transactionStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12347</referenceCode>
<clientCorrelator>55557</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```

5.16.4.4 Example 4: update (release) remaining amount reservation (Informative)

5.16.4.4.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>5</amountReserved>
  </paymentAmount>
  <transactionStatus>Released</transactionStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12348</referenceCode>
  <clientCorrelator>55558</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.4.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>

```

```

<currency>USD</currency>
<amount>5</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountCharged>5</totalAmountCharged>
<amountReserved>0</amountReserved>
</paymentAmount>
<transactionStatus>Released</transactionStatus>
<referenceSequence>3</referenceSequence>
<referenceCode>REF-12348</referenceCode>
<clientCorrelator>55558</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```

5.16.4.5 Example 5: update (reserve) additional amount for amount reservation (Informative)

5.16.4.5.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12349</referenceCode>
  <clientCorrelator>55559</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.5.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12349</referenceCode>
  <clientCorrelator>55559</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```

5.16.4.6 Example 6 update (release) portion of amount reservation (Informative)

5.16.4.6.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in</description>
      <currency>USD</currency>
      <amount>6</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionStatus>Released</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12340</referenceCode>
  <clientCorrelator>55550</clientCorrelator>
</payment:amountReservationTransaction>

```

5.16.4.6.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentAmount>
      <chargingInformation>
        <description>Test amount reservation transaction in</description>
        <currency>USD</currency>
        <amount>6</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>0</totalAmountCharged>
      <amountReserved>9</amountReserved>
    </paymentAmount>
    <transactionStatus>Released</transactionStatus>
    <referenceSequence>2</referenceSequence>
    <referenceCode>REF-12340</referenceCode>
    <clientCorrelator>55550</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
  </payment:amountReservationTransaction>

```

5.16.5 POST

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

5.16.6 DELETE

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

5.17 Resource: All volume reservation and release transactions for an end user

The resource used is:

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

This resource is used to provide access to all the volume reservation and release 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:80/ParlayREST |
| 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 [REST_TS_Common].

5.17.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.17.3 GET

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

Note: No ParlayX SOAP equivalent.

5.17.3.1 Example: get all volume reservation transactions (Informative)

5.17.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.17.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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 IN "RESERVED" STATE -->
  <volumeReservationTransaction>
    <endUserId>tel:+16309700001</endUserId>

    <paymentVolume>
      <billingText>Test volume reservation transaction</billingText>
```

```

<volume>10</volume>
<ratingParameter>
  <name>unit</name>
  <value>minutes</value>
</ratingParameter>
<totalVolumeCharged>0</totalVolumeCharged>
<volumeReserved>10</volumeReserved>
</paymentVolume>
<transactionStatus>Reserved</transactionStatus>
<referenceSequence>1</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>66666</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

</volumeReservationTransaction>
<!--
VOLUME RESERVATION TRANSACTION IN "RESERVED" STATE AFTER ADDITIONAL
VOLUME IS APPLIED
-->
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>15</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66667</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

</volumeReservationTransaction>
<!-- VOLUME RESERVATION TRANSACTION IN "CHARGED" STATE -->
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66668</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

```

```

</volumeReservationTransaction>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/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.

Note: ParlayX SOAP equivalent is ReserveVolume.

5.17.5.1 Example 1: create reserve volume

(Informative)

5.17.5.1.1 Request

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
    <paymentVolume>
      <billingText>Test volume reservation transaction</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
    </paymentVolume>
    <transactionStatus>Reserved</transactionStatus>
    <referenceSequence>1</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>66666</clientCorrelator>
  </payment:volumeReservationTransaction>

```

5.17.5.1.2 Response

```

HTTP/1.1 201 Created

```



```

Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME RESERVATION TRANSACTION -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
</resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

</payment:volumeReservationTransaction>

```

5.17.5.2 Example 2: create reserve volume with invalid endUserId (Informative)

5.17.5.2.1 Request

```

POST ..{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700000</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>

```

```
<clientCorrelator>66667</clientCorrelator>
</payment:volumeReservationTransaction>
```

5.17.5.2.2 Response

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: 12345
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</text>
    <variables>Invalid tel:+016309700000</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 and release transaction for an end user

The resource used is:

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

This resource is used to provide access to an individual volume reservation or release 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:80/ParlayREST |
| 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 [REST_TS_Common].

5.18.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.18.3 GET

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

Note: No ParlayX SOAP equivalent.

5.18.3.1 Example: get volume reservation

(Informative)

5.18.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.18.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
```

```

<referenceCode>REF-12345</referenceCode>
<clientCorrelator>66666</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>

</payment:volumeReservationTransaction>

```

5.18.4 PUT

This operation is used to change transaction state to “charged” or “released” for a volume reservation or release transaction for an end user.

Note: ParlayX SOAP equivalent is ReserveAdditionalVolume, ChargeReservation, ReleaseReservation.

5.18.4.1 Example: update charge volume (Informative)

5.18.4.1.1 Request

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
</payment:volumeReservationTransaction>

```

5.18.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</payment:volumeReservationTransaction>

```

5.18.5 POST

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

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

5.19 Resource: Individual amount for volume reservation or release transaction for an end user

The resource used is:

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

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:80/ParlayREST |
| 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 [REST_TS_Common].

5.19.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

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.

Note: No ParlayX SOAP equivalent.

5.19.3.1 Example: get amount for volume reservation (Informative)

5.19.3.1.1 Request

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.19.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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={transactionId} of endUserId={endUserId}</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`

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: <code>http://example.com:80/ParlayREST</code> |
| 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 [REST_TS_Common].

5.20.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-5].

5.20.3 GET

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

Note: ParlayX SOAP equivalent is GetAmount.

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 ../{apiVersion}/payment/{endUserId}/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MONTHLY-PLAN-001"
```

5.20.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
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={endUserId} 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].

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 Version 1.0 History

| Document Identifier | Date | Sections | Description |
|--|-------------|---|---|
| Draft Versions OMA-TS-ParlayREST-Payment-V1_0 | 1 Dec 2009 | 4.1, 5 Appendix A 5.1 5.15 5.1 5.2 | Added OMA-ARC-REST-2009-0072R01- CR_ParlayREST_Payment_API Added OMA-ARC-REST-2009-0084R02- CR_Parlay_REST_Payment_API Updated from OMA-ARC-REST-2009-0093R01- CR_Equivalent_PX_SOAP_to_Payment_TS Added: OMA-ARC-REST-2009-0105R01- CR_Adding_Examples_ReserveAdditionalAmount_to_Payment_TS.doc Updated: OMA-ARC-REST-2009-0103- CR_DataStructures_SplitCharge_to_Payment_TS Added: OMA-ARC-REST-2009-0089R01- CR_FlowDiagrams_to_Payment_TS |
| | 2 Dec 2009 | 5.2, 5.6, 5.10, 5.12, 5.14 | Added OMA-ARC-REST-2009-0105R01- CR_Adding_Examples_ReserveAdditionalAmount_to_Payment_TS Added OMA-ARC-REST-2009-0110R03- CR_Updates_SplitRefund_to_Payment_TS.doc |
| | 3 Dec 2009 | 4 2 3 App. C 2, App.D | Updated selfURL to resourceURL and added Link Updated from OMA-ARC-REST-2009-0113-INP_SMS_Intro_section. Updated from OMA-ARC-REST-2009-0114- INP_SMS_TS_Reference_Section Updated from OMA-ARC-REST-2009-0115R01- INP_SMS_TS_Section_3 Added from OMA-ARC-REST-2009-0120-CR_Payment_TS_SCR Added from OMA-ARC-REST-2009-0129R01- CR_PX_Profile_vs_REST_Operations_Table_PaymentAPI |
| | 7 Dec 2009 | 5.2 | Added from OMA-ARC-REST-2009-0077- CR_Payment_API_Optionality.doc and added OMA-ARC-REST-2009- 0118R01-CR_Outline_for_request_response_examples |
| | 11 Dec 2009 | all | Update after final CC, see OMA-ARC-REST-2009-0170- MINUTES_11Dec2009_CC for details OMA-ARC-REST-2009-0164-CR_RequestError_Issue_PAY_2 OMA-ARC-REST-2009-0158-CR_Issue_PAY_18 OMA-ARC-REST-2009-0153-CR_Payment_Fix_unbounded_Link OMA-ARC-REST-2009-0150R01- CR_Issue_PAY_6_Missing_XML_Examples OMA-ARC-REST-2009-0146- CR_Issue_PAY_12_Missing_Element_Descriptions |
| | 15 Dec 2009 | all | Added missing 151, and 129 |
| | 16 Dec 2009 | All | Editorial fixes: History table Styles as per template |
| | 26 Jan 2010 | All | CONRR editorial comments applied, D0010, D0015 |
| | 02 Feb 2010 | All | Update OMA-ARC-REST-2010-0021 |
| | 03 Feb 2010 | all | Applied G001, G002, G004 |
| | 04 Feb 2010 | All | Added OMA.ARC-REST-2009-177R01 |
| | 05 Feb 2010 | All | D002 (CR 36R01), D003, D004, D005, D007, D008, D009, D010, D011, D012 D013, D014, D015, D016, D017, D020, D021, D022 D023 (CR 43), D026, D027 |

| Document Identifier | Date | Sections | Description |
|--|-----------------|-------------------------------------|--|
| | 18 Feb 2010 | All | D006 (CR 54R01), D018 (CR 57R01), D019, (CR 65R01), D024 (CR 51R02), D043, editorials |
| | 25 Feb 2010 | All | 82R02. |
| | 05 Mar 2010 | All | CR 91, CR 99R01, CR 111R01 |
| | 07 Mar 2010 | All | Editorials (some created by the editor when splitting tables; others found via pre-wak-thru: endUserId missing as URL parameter in several places SVC002 was replaced with SVC004 in 2 examples, since SVC004 is supposed to be the error message when an incorrect URI is used as part of the URL (no valid addresses provided) |
| | 08 Mar 2010 | All | Editorials (table splits). Also added details on CRs applied in conjunction with CONRR comments (see above rows) |
| | 10th March 2010 | Appendix D 5, Appendix C | JSON examples added New style ('listing') applied to examples |
| | 11 Mar 2010 | All | Applied decisions of Mar 10 walk-thru CC. Exception: removal of the last resource in the resource summary table (and related changes). Needs re-visiting, because that was equivalent to the PX GetAmount. |
| | 16 Mar 2010 | 5.1, 5.3.7, 5.2, 5.20.3, Appendix C | In 5.1 and 5.3.7 edited the Amount Converted resource to show all possible optional parameter. 5.2. changed "Element name" to "Element", "Element type" to "Type" in some table headings for consistency. In 5.20.3, changed all parameters from mandatory to optional (except "volume"); added missed "service" and "operation" to the possible optional parameters. Removed "currency" as an optional parameter, fully aligning the operation with GetAmount equivalent in ParlayX. In Appendix C: Added Type/Values columns; changed "Parameter" to "Name"; added a note to transactionStatus; |
| | 22 Mar 2010 | 5, 5.3 | Implemented CR 134R01, editorials |
| | 27 Mar 2010 | As per CRs | Implemented CR 144, 154R02 (using 137R02 agreed "style" for describing resource & URL parameters), 158R01, 160, editorials |
| Candidate Version: OMA-TS-ParlayREST_Payment-V1_0 | 27 Apr 2010 | All | Status changed to Candidate by TP: OMA-TP-2010-0186- INP_ParlayREST_V1_0_ERP_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 Enabler | 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 A | |

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-O | Support for access to all amount charge and refund transactions for an end-user | 5.5 A.1 | PARLAYREST-PAY-AMNT-TRANS-S-002-O |
| 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 | 5.5.5 A.1 | |
| PARLAYREST-PAY-AMNT-TRANS-S-004-M | Create a refund amount transaction for an end-user - POST | 5.5.5 A.2 | |

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

| Item | Function | Reference | Requirement |
|-----------------|-----------------------|-----------|-----------------------|
| PARLAYREST-PAY- | Support for access to | 5.6 | PARLAYREST-PAY-SPLIT- |

| Item | Function | Reference | Requirement |
|------------------------------------|---|-----------|---------------|
| SPLIT-TRANS-S-001-O | all amount split charge and refund transactions for an end-user | | 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 AND PARLAYREST-PAY-IND-AMNT-S-003-O |
| PARLAYREST-PAY-IND-AMNT-S-002-O | Retrieve individual completed or pending amount transaction - GET | 5.7.3 | |
| PARLAYREST-PAY-IND-AMNT-S-003-O | Update individual completed or pending amount transaction - PUT | 5.7.4 | |

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 AND PARLAYREST-PAY-IND-SPLIT-S-003-O |
| PARLAYREST-PAY-IND-SPLIT-S-002-O | Retrieve individual completed or pending split amount transaction - GET | 5.8.3 | |
| PARLAYREST-PAY-IND-SPLIT-S-003-O | Update individual completed or pending split amount transaction - PUT | 5.8.4 | |

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- | 5.9.5 | |

| Item | Function | Reference | Requirement |
|----------------------------------|---|-----------|-------------|
| | user - POST | | |
| 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 AND PARLAYREST-PAY-IND-VOL-S-003-O |
| PARLAYREST-PAY-IND-VOL-S-002-O | Retrieve individual completed or pending transaction - GET | 5.11.3 | |
| PARLAYREST-PAY-IND-VOL-S-003-O | Update individual completed or pending volume transaction - PUT | 5.11.4 | |

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 AND PARLAYREST-PAY-IND-SPLIT-VOL-S-003-O |
| PARLAYREST-PAY-IND-SPLIT-VOL-S-002-O | Retrieve individual completed or pending split volume charge transaction - GET | 5.12.3 | |
| PARLAYREST-PAY-IND-SPLIT-VOL-S-003-O | Update individual completed or pending split volume transaction - PUT | 5.12.4 | |

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-O | Support for access to all completed and pending amount reservation and release transactions | 5.15 A.3 | PARLAYREST-PAY-AMNT-RES-TRANS-S-002-O |
| 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 transaction for an end-user - POST | 5.15.6 A.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-O | Support for access to individual completed and pending amount reservation and release transaction | 5.16 A.4 A.5 | PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-002-O |

| Item | Function | Reference | Requirement |
|---|--|----------------------|-------------|
| PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-002-O | Obtain all transactions for an end-user - GET | 5.16.4 | |
| P PARLAYREST-PAY-IND-AMNT-RES-TRANS-S-003-M | Create a new transaction for an end-user - PUT | 5.16.5 A.4 A.5 | |

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 and release 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-003-O | Create a new transaction for an end-user - POST | 5.17.6 | |

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 and release 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 | Create a new transaction for an end-user - PUT | 5.18.5 | |

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.Converted Server

| Item | Function | Reference | Requirement |
|------|----------|-----------|-------------|
|------|----------|-----------|-------------|

| 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

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.

Note: ParlayX SOAP equivalent is ChargeAmount.

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:+447990123456. If an Anonymous Customer Reference (ACR) is available, include the protocol prefix 'acr:' followed by the ACR. |
| transactionStatus | xsd:string | No | e.g. charged, refunded, etc (see TransactionStatus 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 | No | The amount to be charged |
| 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 | <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. 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 identify the product being purchased. |

C.1.1 Example

(Informative)

C.1.1.1 Request

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com:80
```

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

```
Content-Length: 12345
```

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

```
endUserId=tel:%2B16309700001&
```

```
transactionStatus=charged&
```

```
description= Test%20amount%20transaction%20in%20charged%20state&
```

```
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/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "charged" state</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>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</payment:amountTransaction>
```

C.2 Refund an Amount

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

Note: ParlayX SOAP equivalent is RefundAmount.

The request parameters are as follows:

| Name | Type/Values | Optional | Description |
|----------------------|-------------------|----------|---|
| endUserId | xsd:anyURI | No | Unique identifier for the end user's account |
| transactionStatus | TransactionStatus | No | e.g. charged, refunded, etc (see TransactionStatus 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 | No | The amount to be charged |
| 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 | <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 indentify the |

| | | | |
|--|--|--|--------------------------|
| | | | product being purchased. |
|--|--|--|--------------------------|

C.2.1 Example

(Informative)

C.2.1.1 Request

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com:80
```

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

```
Content-Length: 12345
```

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

```
endUserId=tel:%2B16309700001&
transactionStatus=refunded&
description= Test%20amount%20transaction%20in%20refunded%20state&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.2.1.2 Response

```
HTTP/1.1 200 OK
```

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

```
Content-Length: 12345
```

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

```
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
```

```

<onBehalfOf>Example Games Inc</onBehalfOf>
<purchaseCategoryCode>Game</purchaseCategoryCode>
<channel>WAP</channel>
<taxAmount>0</taxAmount>
</chargingMetaData>
</paymentAmount>
<transactionStatus>Refunded</transactionStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</payment:amountTransaction>

```

C.3 Reserve an Amount

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

Note: ParlayX SOAP equivalent is ReserveAmount.

The request parameters are as follows:

| Name | Type/Values | Optional | Description |
|-------------------|-------------------|----------|---|
| endUserId | xsd:anyURI | No | Unique identifier for the end user's account |
| transactionStatus | TransactionStatus | No | e.g. charged, reserved, etc (see TransactionStatus 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 | No | The amount to be charged |
| 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 | 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: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 ../{apiVersion}/payment/{endUserId}/transactions/amountReservation HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com:80
```

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

```
Content-Length: 12345
```

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

```
endUserId=tel:%2B16309700001&
```

```
transactionStatus=reserved&
```

```
description= Test%20amount%20transaction%20in%reserved%20state&
```

```
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/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction
  xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>

  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in "charged" state</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>
  <transactionStatus>Reserved</transactionStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>TEST-012345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>
```

C.4 Reserve an Additional Amount

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

Note: ParlayX SOAP equivalent is ReserveAdditionalAmount.

The request parameters are as follows:

| Name | Type/Values | Optional | Description |
|------|-------------|----------|-------------|
|------|-------------|----------|-------------|

| | | | |
|----------------------|-------------------|-----|---|
| endUserId | xsd:anyURI | No | Unique identifier for the end user's account |
| transactionStatus | TransactionStatus | No | e.g. charged, reserved, etc (see TransactionStatus 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 | No | The amount to be charged |
| 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 | <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> |
| 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 |
| 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

```
PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
endUserId=tel:%2B16309700001&
transactionStatus=reserved&
description= Test%20amount%20transaction%20in%20reserved%20additional%20state&
currency=USD&
amount=5&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
referenceSequence=2&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction in "charged" state</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST-012345</code>
```

```

</chargingInformation>
<totalAmountCharged>0</totalAmountCharged>
<amountReserved>15</amountReserved>
<chargingMetaData>
  <onBehalfOf>Example Games Inc</onBehalfOf>
  <purchaseCategoryCode>Game</purchaseCategoryCode>
  <channel>WAP</channel>
  <taxAmount>0</taxAmount>
</chargingMetaData>
</paymentAmount>
<transactionStatus>Reserved</transactionStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```

C.5 Charge to a Reservation

This operation is used to charge against an existing reservation.

Note: ParlayX SOAP equivalent is ChargeReservation.

The request parameters are as follows:

| Name | Type/Values | Optional | Description |
|-------------------|-------------------|----------|--|
| endUserId | xsd:anyURI | No | Unique identifier for the end user's account |
| transactionStatus | TransactionStatus | No | e.g. charged, reserved, etc (see TransactionStatus 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 | No | The amount to be charged |
| 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 | <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</p> |

| | | | |
|----------------------|-------------|-----|---|
| | | | representation of this resource. In case the field is not present, the server SHALL NOT generate it. |
| referenceCode | xsd:string | Yes | 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 indentify the product being purchased. |

C.5.1 Example

(Informative)

C.5.1.1 Request

```
PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```

endUserId=tel:%2B16309700001&
transactionStatus=charged&
description= Test%20amount%20transaction%20in%20refunded%20state&
currency=USD&
amount=5&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
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: 12345
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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>10</amountReserved>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{{serverRoot}}/{{apiVersion}}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

```

C.6 Release a Reservation

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

Note: ParlayX SOAP equivalent is ReleaseReservation

The request parameters are as follows:

| Name | Type/Values | Optional | Description |
|----------------------|-------------------|----------|--|
| endUserId | xsd:anyURI | No | Unique identifier for the end user's account |
| transactionStatus | TransactionStatus | No | e.g. charged, reserved, etc (see TransactionStatus 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 | No | The amount to be charged |
| code | xsd:string | Yes | Charging code referencing a contract under which the charge is applied |
| 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). |
| referenceCode | xsd:string | Yes | 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 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. |
| 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

```
PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
```

```
Accept: application/xml
```

```
Host: example.com:80
```

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

```
Content-Length: 12345
```

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

```
endUserId=tel:%2B16309700001&
transactionStatus=released&
description= Test%20amount%20transaction%20in%20released%20state&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
referenceSequence=4&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP
```

C.6.1.2 Response

```
HTTP/1.1 200 OK
```

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

```
Content-Length: 12345
```


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:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>0</amountReserved>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
    </chargingMetaData>
  </paymentAmount>
  <transactionStatus>Released</transactionStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation{transactionId}</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 or response for various operations using a JSON binding. The examples follow the XML to JSON serialization guidelines in [REST_WP]. A JSON response may be obtained by following the content negotiation guidelines section of [REST_WP].

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

D.1.1.1 Get all transactions (section 5.4.3.1.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountReservationTransaction": [
    {
      "clientCorrelator": "55555",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
          "amount": "15",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount reservation transaction in \"Charged\" state"
        }
      },
      "totalAmountCharged": "25"
    },
    {
      "referenceCode": "REF-12345",
      "referenceSequence": "2",
      "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
      "transactionStatus": "Charged"
    }
  ],
  {
    "clientCorrelator": "55556",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "amountReserved": "10",
      "chargingInformation": {
```

```

        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test reserve amount"
    },
    "totalAmountCharged": "0"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionStatus": "Reserved"
}
],
"amountSplitTransaction": {
    "clientCorrelator": "54323",
    "endUserShare": [
        {
            "endUserId": "tel:+16309700001",
            "percent": "20"
        },
        {
            "endUserId": "tel:+16309700002",
            "percent": "80"
        }
    ],
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction in \"Charged\" state"
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
    "transactionStatus": "Charged"
},
"amountTransaction": [
    {
        "clientCorrelator": "54321",
        "endUserId": "tel:+16309700001",
        "paymentAmount": {
            "chargingInformation": {
                "amount": "10",
                "code": "TEST-012345",
                "currency": "USD",
                "description": "Test amount transaction in \"Charged\" state"
            },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",

```

```

    "transactionStatus": "Charged"
  },
  {
    "clientCorrelator": "54322",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Refunded\" state"
      },
      "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL":
"http://(serverRoot)/(apiVersion)/payment/{endUserId}/transactions/amount/{transactionId}",
    "transactionStatus": "Refunded"
  }
],
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions",
"volumeReservationTransaction": [
  {
    "clientCorrelator": "66666",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction in \"Reserved\" state",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "0",
      "volume": "10",
      "volumeReserved": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "resourceURL":
"http://(serverRoot)/(apiVersion)/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
    "transactionStatus": "Reserved"
  },
  {
    "clientCorrelator": "66667",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction in \"Charged\" state",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "25",
      "volume": "10",
      "volumeReserved": "0"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "4",

```

```

    "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
    "transactionStatus": "Charged"
  }
],
"volumeSplitTransaction": {
  "clientCorrelator": "54325",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
  "transactionStatus": "Charged"
},
"volumeTransaction": [
  {
    "clientCorrelator": "55551",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume transaction in \"Charge\" state",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL":
"http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
    "transactionStatus": "Charged"
  },
  {
    "clientCorrelator": "54324",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume transaction in \"Refunded\" state",
      "ratingParameter": {
        "name": "unit",

```

```

        "value": "minutes"
      },
      "totalVolumeRefunded": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL":
"http://{{serverRoot}}/{{apiVersion}}/payment/{{endUserId}}/transactions/volume/{{transactionId}}",
    "transactionStatus": "Refunded"
  }
]
}}

```

D.1.1.2 Request with invalid endUserId (section 5.4.3.2)

Request:

```

GET ../{{apiVersion}}/payment/{{endUserId}}/transactions HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {
  "link": {
    "href": "http://{{serverRoot}}/apiVersion}/payment/{{endUserId}}/transactions",
    "rel": "PaymentTransactionList"
  },
  "serviceException": {
    "messageId": "SVC0004",
    "text": "No valid addresses provided in message part %1",
    "variables": "tel:+016309700000"
  }
}}

```

D.1.1.3 Get all amount transactions (section 5.5.3.1)

Request:

```

GET ../{{apiVersion}}/payment/{{endUserId}}/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```

{"paymentTransactionList": {
  "amountTransaction": [
    {
      "clientCorrelator": "54321",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST-012345",
          "currency": "USD",
          "description": "Test amount transaction in \"Charged\" state"
        },
        "totalAmountCharged": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
      "transactionStatus": "Charged"
    },
    {
      "clientCorrelator": "54321",
      "endUserId": "tel:1234567890",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount transaction in \"Charged\" state"
        },
        "totalAmountCharged": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
      "transactionStatus": "Charged"
    },
    {
      "clientCorrelator": "54330",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount transaction in \"Refunded\" state"
        },
        "totalAmountRefunded": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
      "transactionStatus": "Refunded"
    }
  ],
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount"
}}

```

D.1.1.4 Create charge amount (section 5.5.5.1)

Request:

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Charged\" state"
      }
    }
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Charged"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Charged\" state"
      }
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
  "transactionStatus": "Charged"
}
```

D.1.1.5 Create refund amount (section 5.5.5.2)

Request:

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/json
```



```
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Refunded\" state"
      }
    },
    "referenceCode": "REF-12345",
    "transactionStatus": "Refunded"
  }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Refunded\" state"
      }
    },
    "totalAmountRefunded": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
  "transactionStatus": "Refunded"
}
```

D.1.1.6 Client retries POST with same clientCorrelator (section 5.5.5.3)

Request:

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": { "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction in \"Charged\" state"
    }
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Charged"
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}
{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction in \"Charged\" state"
    }
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
"transactionStatus": "Charged"
}
```

D.1.1.7 Get all amount split transactions (section 5.6.3.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amountSplit HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```

{"paymentTransactionList": {
  "amountSplitTransaction": [
    {
      "clientCorrelator": "55552",
      "endUserShare": [
        {
          "endUserId": "tel:+16309700001",
          "percent": "30"
        },
        {
          "endUserId": "tel:+16309700001",
          "percent": "70"
        }
      ]
    },
    {
      "paymentAmount": {
        "chargingInformation": {
          "amount": "20",
          "code": "TEST-012345",
          "currency": "USD",
          "description": "Test amount transaction in \"Charged\" state"
        },
        "totalAmountCharged": "20"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
      "transactionStatus": "Charged"
    }
  ],
  {
    "clientCorrelator": "55553",
    "endUserShare": [
      {
        "endUserId": "tel:+16309700001",
        "percent": "20"
      },
      {
        "endUserId": "tel:+16309700002",
        "percent": "80"
      }
    ],
    {
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST-012346",
          "currency": "USD",
          "description": "Test amount transaction in \"Charged\" state"
        },
        "totalAmountCharged": "10"
      },
      "referenceCode": "REF-12346",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
      "transactionStatus": "Charged"
    }
  ]
},
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit"
}}

```

D.1.1.8 Create split charge amount (section 5.6.5.1)

Request:

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amountSplit HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{
  "amountSplitTransaction": {
    "clientCorrelator": "54431",
    "endUserShare": [
      {
        "endUserId": "tel:+16309700001",
        "percent": "30"
      },
      {
        "endUserId": "tel:+16309700001",
        "percent": "70"
      }
    ],
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Charged\" state"
      }
    },
    "referenceCode": "REF-12345",
    "transactionStatus": "Charged"
  }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}
```

```
{
  "amountSplitTransaction": {
    "clientCorrelator": "54431",
    "endUserShare": [
      {
        "endUserId": "tel:+16309700001",
        "percent": "30"
      },
      {
        "endUserId": "tel:+16309700002",
        "percent": "70"
      }
    ],
  }
}
```

```

"paymentAmount": {
  "chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction in \"Charged\" state"
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
"transactionStatus": "Charged"
}}

```

D.1.1.9 Get amount charge (Section 5.7.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}?resFormat=XML HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction in \"Charged\" state"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.10 Update amount transaction (section 5.7.4.1)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId} HTTP/1.1
Accept: application/json

```

Host: example.com:80
 Content-Type: application/json
 Content-Length: 12345
 Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Refunded\" state"
      },
      "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "transactionStatus": "Refunded"
  }
}
```

Response:

HTTP/1.1 200 OK
 Content-Type: application/json
 Content-Length: 12345
 Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
      "chargingInformation": {
        "amount": "10",
        "code": "TEST-012345",
        "currency": "USD",
        "description": "Test amount transaction in \"Refunded\" state"
      },
      "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
    "transactionStatus": "Refunded"
  }
}
```

D.1.1.11 Get amount split charge (section 5.8.3.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}
HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
{"amountSplitTransaction": {
  "clientCorrelator": "54321",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "30"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "70"
    }
  ],
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction in \"Charged\" state"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.12 Update split charge amount transaction (section 5.8.4.1)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountSplitTransaction": {
  "clientCorrelator": "54321",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "30"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "70"
    }
  ],

```

```

"paymentAmount": {
  "chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount transaction in \"Charged\" state"
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
{"amountSplitTransaction": {
  "clientCorrelator": "54321",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "30"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "70"
    }
  ],
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction in \"Charged\" state"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.13 Get all volume charge and refund transactions (section 5.9.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:


```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume",
  "volumeTransaction": [
    {
      "clientCorrelator": "55551",
      "endUserId": "tel:+16309700001",
      "paymentVolume": {
        "billingText": "Test volume transaction in \"Charged\" state",
        "ratingParameter": {
          "name": "unit",
          "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
      "transactionStatus": "Charged"
    },
    {
      "clientCorrelator": "55552",
      "endUserId": "tel:+16309700001",
      "paymentVolume": {
        "billingText": "Test volume transaction in \"Refunded\" state",
        "ratingParameter": {
          "name": "unit",
          "value": "minutes"
        },
        "totalVolumeRefunded": "10",
        "volume": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
      "transactionStatus": "Refunded"
    }
  ]
}
}
}

```

D.1.1.14 Create charge volume, returning a representation of created resource (section 5.9.5.1)

Request:

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json

```

```

{"volumeTransaction": {

```

```

"clientCorrelator": "55551",
"endUserId": "tel:+16309700001",
"paymentVolume": {
  "billingText": "Test volume transaction in \"Charged\" state",
  "ratingParameter": {
    "name": "unit",
    "value": "minutes"
  },
  "volume": "10"
},
"referenceCode": "REF-12345",
"transactionStatus": "Charged"
}}

```

Response:

```

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

{"volumeTransaction": {
  "clientCorrelator": "55551",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.15 Create charge volume, returning the location of created resource (section 5.9.5.2)

Request:

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json

{"volumeTransaction": {
  "clientCorrelator": "55551",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",

```

```

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

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

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

```

D.1.1.16 Create refund volume (section 5.9.5.2.3)

Request:

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml

{"volumeTransaction": {
  "clientCorrelator": "55552",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Refunded\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Refunded"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}

{"volumeTransaction": {
  "clientCorrelator": "55552",
  "endUserId": "tel:+16309700001",

```

```

"paymentVolume": {
  "billingText": "Test volume transaction in \"Refunded\" state",
  "ratingParameter": {
    "name": "unit",
    "value": "minutes"
  },
  "totalVolumeRefunded": "10",
  "volume": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
"transactionStatus": "Refunded"
}}

```

D.1.1.17 Get all volume split charge transactions (section 5.10.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit
HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit",
  "volumeSplitTransaction": {
    "clientCorrelator": "55553",
    "endUserShare": [
      {
        "endUserId": "tel:+16309700001",
        "percent": "20"
      },
      {
        "endUserId": "tel:+16309700002",
        "percent": "80"
      }
    ],
    "paymentVolume": {
      "billingText": "Test volume transaction in \"Charged\" state",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
  }
}

```

```

    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
    "transactionStatus": "Charged"
  }
}

```

D.1.1.18 Create volume split charge (section 5.10.5.1.)

Request:

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

```

```

{"volumeSplitTransaction": {
  "clientCorrelator": "55553",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}

{"volumeSplitTransaction": {
  "clientCorrelator": "55553",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    }
  ]
}

```

```

    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.19 Get volume charge (section 5.11.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.20 Update volume charge transaction (section 5.11.4.1)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.21 Get volume split charge (section 5-12-3-1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeSplitTransaction": {
  "clientCorrelator": "55556",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
  "transactionStatus": "Charged"
}}
```

D.1.1.22 Update volume split charge transaction (section 5.12.4.1)

Request:

```
PUT ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeSplitTransaction": {
  "clientCorrelator": "55556",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    }
  ]
}}
```



```

    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeSplitTransaction": {
  "clientCorrelator": "55556",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.23 Get amount for volume charge (section 5.13.3.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/volume/{transactionId}/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeSplitTransaction": {
  "clientCorrelator": "55556",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction in \"Charged\" state",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
  "transactionStatus": "Charged"
}}
```

D.1.1.24 Get amount for volume split charge (section 5.14.3.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
```

```

Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

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

```

D.1.1.25 Get all amount reservation transactions (section 5.15.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/amountReservation HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "amountReservationTransaction": [
    {
      "clientCorrelator": "55555",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
          "amount": "15",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount reservation transaction in \"Charged\" state"
        },
        "totalAmountCharged": "25"
      },
      "referenceCode": "REF-12345",
      "referenceSequence": "2",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
      "transactionStatus": "Charged"
    },
    {
      "clientCorrelator": "55556",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "amountReserved": "10",
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",

```

```

        "currency": "USD",
        "description": "Test reserve amount"
    },
    "totalAmountCharged": "0"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionStatus": "Reserved"
},
{
    "clientCorrelator": "55557",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
        "amountReserved": "15",
        "chargingInformation": {
            "amount": "15",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test reserve amount"
        }
    },
    "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionStatus": "Reserved"
},
{
    "clientCorrelator": "55557",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {
        "amountReserved": "25",
        "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test reserve amount"
        }
    },
    "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"referenceSequence": "3",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionStatus": "Reserved"
}
},
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation"
}}

```

D.1.1.26 Create reserve amount (section 5.15.5.1)

Request:

```
POST ../{apiVersion}/payment/{endUserId}/transactions/amountReservation HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount reservation transaction"
  }},
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "transactionStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}
```

```
{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {
    "amount": "10",
    "code": "TEST-012345",
    "currency": "USD",
    "description": "Test amount reservation transaction"
  }},
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "transactionStatus": "Reserved"
}}
```

D.1.1.27 Get amount reservation (section 5.16.3.1)

Request:

```
GET ../{apiVersion}/{endUserId}/transactions/amountReservation/{transactionId}
HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Reserved"
}}

```

D.1.1.28 Update (charge) amount for amount reservation (section 5.16.4.1)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55555",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.29 Update (release) amount for amount reservation

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55556",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12346",
  "referenceSequence": "2",
  "transactionStatus": "Released"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55556",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12346",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Released"
}}

```

D.1.1.30 Update (charge partial) amount for amount reservation (section 5.16.4.3)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55557",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12347",
  "referenceSequence": "2",
  "transactionStatus": "Charged"
}}

```

Response:


```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55557",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "5",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "5"
  },
  "referenceCode": "REF-12347",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.31 Update (release) remaining amount reservation (section 5.16.4.4)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55558",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "5",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "5"
  },
  "referenceCode": "REF-12348",
  "referenceSequence": "3",
  "transactionStatus": "Released"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55558",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction"
    },
    "totalAmountCharged": "5"
  },
  "referenceCode": "REF-12348",
  "referenceSequence": "3",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Released"
}}

```

D.1.1.32 Update (reserve) additional amount for amount reservation (section 5.16.4.5)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55559",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
      "amount": "5",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction in"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12349",
  "referenceSequence": "2",
  "transactionStatus": "Reserved"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55559",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "15",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction in"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12349",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Reserved"
}}

```

D.1.1.33 Update (release) portion of amount reservation (section 5.16.4.6)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55550",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "15",
    "chargingInformation": {
      "amount": "6",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction in"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12340",
  "referenceSequence": "2",
  "transactionStatus": "Released"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"amountReservationTransaction": {
  "clientCorrelator": "55550",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "amountReserved": "9",
    "chargingInformation": {
      "amount": "6",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction in"
    },
    "totalAmountCharged": "0"
  },
  "referenceCode": "REF-12340",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionStatus": "Released"
}}

```

D.1.1.34 Get all volume reservation transactions (section 5.17.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentTransactionList": {
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation",
  "volumeReservationTransaction": [
    {
      "clientCorrelator": "66666",
      "endUserId": "tel:+16309700001",
      "paymentVolume": {
        "billingText": "Test volume reservation transaction",
        "ratingParameter": {
          "name": "unit",
          "value": "minutes"
        }
      },
      "totalVolumeCharged": "0",
      "volume": "10",
    }
  ]
}

```

```

    "volumeReserved": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
  "transactionStatus": "Reserved"
},
{
  "clientCorrelator": "66667",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "15",
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "3",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
  "transactionStatus": "Reserved"
},
{
  "clientCorrelator": "66668",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "25",
    "volume": "10",
    "volumeReserved": "0"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "4",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
  "transactionStatus": "Charged"
}
]
}}

```

D.1.1.35 Create reserve volume (section 5.17.5.1)

Request:

```

POST ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json

```

Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "volumeReservationTransaction": {
    "clientCorrelator": "66666",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "transactionStatus": "Reserved"
  }
}
```

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}

```
{
  "volumeReservationTransaction": {
    "clientCorrelator": "66667",
    "endUserId": "tel:+16309700000",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "transactionStatus": "Reserved"
  }
}
```

D.1.1.36 Create reserve volume with invalid endUserId (section 5.17.5.2)

Request:

POST ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

```
{
  "volumeReservationTransaction": {
    "clientCorrelator": "66667",
    "endUserId": "tel:+16309700000",
    "paymentVolume": {
      "billingText": "Test volume reservation transaction",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "transactionStatus": "Reserved"
  }
}
```

Response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC0002",
  "text": "Invalid input value",
  "variables": "Invalid tel:+016309700000"
}}}

```

D.1.1.37 Get volume reservation (section 5.18.3.1)

Request:

```
GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId} HTTP/1.1
Accept: application/xml
Host: example.com:80

```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    }
  }
}
```

```

    },
    "totalVolumeCharged": "0",
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
  "transactionStatus": "Reserved"
}}

```

D.1.1.38 Update charge volume (section 5.18.4.1)

Request:

```

PUT ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId} HTTP/1.1
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionStatus": "Charged"
}}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"volumeReservationTransaction": {
  "clientCorrelator": "66666",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    }
  }
}}

```



```

    },
    "totalVolumeCharged": "10",
    "volume": "10",
    "volumeReserved": "0"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
  "transactionStatus": "Charged"
}}

```

D.1.1.39 Get amount for volume reservation (section 5.19.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "10 Minutes converted to USD for transaction={transactionId} of endUserId={endUserId}"
}}}

```

D.1.1.40 Get amount converted from volume (section 5.20.3.1)

Request:

```

GET ../{apiVersion}/payment/{endUserId}/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MONTHLY-PLAN-001"&currency="USD"
Accept: application/json
Host: example.com:80

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"paymentAmount": {"chargingInformation": {
  "amount": "10",
  "currency": "USD",
  "description": "100 Minutes converted to USD for endUserId={endUserId} according to contract=\"MONTHLY-PLAN-001\""
}}}

```

}}