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 ............................................................................................... 21
      5.2.1 Type: PaymentTransactionList ..................................................................................................................... 21
      5.2.2 Type: AmountTransaction .......................................................................................................................... 21
      5.2.3 Type: AmountSplitTransaction .................................................................................................................. 23
      5.2.4 Type: VolumeTransaction ........................................................................................................................... 24
      5.2.5 Type: VolumeSplitTransaction ................................................................................................................... 26
      5.2.6 Type: AmountReservationTransaction ..................................................................................................... 27
      5.2.7 Type: VolumeReservationTransaction ....................................................................................................... 29
      5.2.8 Type: PaymentAmount .................................................................................................................................. 31
      5.2.9 Type: PaymentVolume .................................................................................................................................. 31
      5.2.10 Type: ChargingMetaData ........................................................................................................................... 32
      5.2.11 Type: EndUserShare ................................................................................................................................... 33
      5.2.12 Type: RatingParameter .............................................................................................................................. 33
      5.2.13 Enumeration: TransactionOperationStatus ............................................................................................... 33
      5.2.14 Values of the Link “rel” attribute ................................................................................................................ 34
   5.3 SEQUENCE DIAGRAMS ............................................................................................................................................ 34
      5.3.1 Amount charge and refund transaction .......................................................................................................... 34
      5.3.2 Amount split charge transaction .................................................................................................................... 35
      5.3.3 Volume charge and refund transaction .......................................................................................................... 35
      5.3.4 Volume split charge transaction .................................................................................................................... 36
      5.3.5 Amount reservation transaction .................................................................................................................... 37
      5.3.6 Volume reservation transaction ..................................................................................................................... 38
      5.3.7 Amount converted from volume transaction ................................................................................................ 40
   5.4 RESOURCE: ALL PAYMENT TRANSACTIONS FOR AN END USER ............................................................................... 40
      5.4.1 Request URI variables .................................................................................................................................. 40
      5.4.2 Response Codes .............................................................................................................................................. 41
         5.4.2.1 HTTP Response Codes .......................................................................................................................... 41
         5.4.2.2 Exception fault codes ............................................................................................................................ 41
      5.4.3 GET ............................................................................................................................................................... 41
         5.4.3.1 Example 1: get all transactions (Informative) ........................................................................................ 41
            5.4.3.1.1 Request ........................................................................................................................................... 41
            5.4.3.1.2 Response ....................................................................................................................................... 41
         5.4.3.2 Example 2: request with invalid (non-existing) endUserId (Informative) ............................................. 45
            5.4.3.2.1 Request ....................................................................................................................................... 45
            5.4.3.2.2 Response ..................................................................................................................................... 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
5.8.3.1.2 Request ................................................................. 59
5.8.3.1.3 Response ............................................................ 59
5.8.3.2 Request ................................................................. 59
5.8.3.2.2 Response .......................................................... 59
5.8.4 PUT .............................................................................. 59
5.8.5 POST ............................................................................. 59
5.8.6 DELETE ........................................................................ 60
5.9 RESOURCE: ALL VOLUME CHARGE AND REFUND TRANSACTIONS FOR AN END USER ......................................................... 60

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document

[OMA-Template-Spec-20090101-I]
5.9.1 Request URI variables ................................................................................................................................. 60
5.9.2 Response Codes .............................................................................................................................................. 60
  5.9.2.1 HTTP Response Codes .......................................................................................................................... 60
  5.9.2.2 Exception fault codes .......................................................................................................................... 60
5.9.3 GET ................................................................................................................................................................. 60
  5.9.3.1 Example: get all volume charge and refund transactions (Informative) ..................................................... 60
    5.9.3.1.1 Request ........................................................................................................................................... 60
    5.9.3.1.2 Response ....................................................................................................................................... 60
5.9.4 PUT ................................................................................................................................................................. 61
5.9.5 POST ............................................................................................................................................................... 62
  5.9.5.1 Example 1: create charge volume, returning a representation of created resource (Informative) ................. 62
    5.9.5.1.1 Request ........................................................................................................................................... 62
    5.9.5.1.2 Response ....................................................................................................................................... 62
  5.9.5.2 Example 2: create charge volume, returning the location of created resource (Informative) .................. 63
    5.9.5.2.1 Request ........................................................................................................................................... 63
    5.9.5.2.2 Response ....................................................................................................................................... 63
  5.9.5.3 Example 3: create refund volume (Informative) ....................................................................................... 63
    5.9.5.3.1 Request ........................................................................................................................................... 63
    5.9.5.3.2 Response ....................................................................................................................................... 64
5.9.6 DELETE ............................................................................................................................................................ 65
5.10 RESOURCE: ALL VOLUME SPLIT CHARGE TRANSACTIONS FOR AN END USER ........................................ 65
  5.10.1 Request URI variables ............................................................................................................................... 65
  5.10.2 Response Codes ........................................................................................................................................... 65
    5.10.2.1 HTTP Response Codes .................................................................................................................... 65
    5.10.2.2 Exception fault codes .................................................................................................................... 65
  5.10.3 GET ............................................................................................................................................................... 65
    5.10.3.1 Example: get all volume split charge transactions (Informative) .................................................... 65
      5.10.3.1.1 Request ....................................................................................................................................... 65
      5.10.3.1.2 Response ................................................................................................................................... 66
  5.10.4 PUT ............................................................................................................................................................... 66
  5.10.5 POST .............................................................................................................................................................. 66
    5.10.5.1 Example: create volume split charge (Informative) ........................................................................ 66
      5.10.5.1.1 Request ....................................................................................................................................... 66
      5.10.5.1.2 Response ................................................................................................................................... 67
  5.10.6 DELETE ......................................................................................................................................................... 68
5.11 RESOURCE: INDIVIDUAL VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER ..................... 68
  5.11.1 Request URI variables .................................................................................................................................. 68
  5.11.2 Response Codes ........................................................................................................................................... 68
    5.11.2.1 HTTP Response Codes .................................................................................................................... 68
    5.11.2.2 Exception fault codes .................................................................................................................... 68
  5.11.3 GET ............................................................................................................................................................... 68
    5.11.3.1 Example: get volume charge (Informative) .................................................................................... 69
      5.11.3.1.1 Request ....................................................................................................................................... 69
      5.11.3.1.2 Response ................................................................................................................................... 69
  5.11.4 PUT ............................................................................................................................................................... 69
  5.11.5 POST .............................................................................................................................................................. 69
  5.11.6 DELETE ........................................................................................................................................................ 69
5.12 RESOURCE: INDIVIDUAL VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER .................................. 70
  5.12.1 Request URI variables .................................................................................................................................. 70
  5.12.2 Response Codes ........................................................................................................................................... 70
    5.12.2.1 HTTP Response Codes .................................................................................................................... 70
    5.12.2.2 Exception fault codes .................................................................................................................... 70
  5.12.3 GET ............................................................................................................................................................... 70
    5.12.3.1 Example: get volume split charge (Informative) ........................................................................ 70
      5.12.3.1.1 Request ....................................................................................................................................... 70
      5.12.3.1.2 Response ................................................................................................................................... 70
  5.12.4 PUT ............................................................................................................................................................... 71
  5.12.5 POST .............................................................................................................................................................. 71
  5.12.6 DELETE ........................................................................................................................................................ 71
5.13 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME CHARGE OR REFUND TRANSACTION FOR AN END USER .... 71
5.13.1 Request URI variables ................................................................. 72
5.13.2 Response Codes ........................................................................ 72
  5.13.2.1 HTTP Response Codes ....................................................... 72
  5.13.2.2 Exception fault codes ......................................................... 72
5.13.3 GET.................................................................................................. 72
  5.13.3.1 Example: get amount for volume charge (Informative) ...... 72
    5.13.3.1.1 Request ................................................................. 72
    5.13.3.1.2 Response .............................................................. 72
  5.13.4 PUT.................................................................................................. 73
  5.13.5 POST.............................................................................................. 73
5.13.6 DELETE............................................................................................ 73

5.14 RESOURCE: INDIVIDUAL AMOUNT FOR VOLUME SPLIT CHARGE TRANSACTION FOR AN END USER ............. 73
  5.14.1 Request URI variables ................................................................. 73
  5.14.2 Response Codes ........................................................................ 73
    5.14.2.1 HTTP Response Codes ................................................... 73
    5.14.2.2 Exception fault codes ....................................................... 73
  5.14.3 GET.................................................................................................. 74
    5.14.3.1 Example: get amount for volume split charge (Informative)
                    ................................................................................... 74
      5.14.3.1.1 Request ................................................................. 74
      5.14.3.1.2 Response .............................................................. 74
  5.14.4 PUT.................................................................................................. 74
  5.14.5 POST.............................................................................................. 74
5.14.6 DELETE............................................................................................ 74

5.15 RESOURCE: ALL AMOUNT RESERVATION TRANSACTIONS FOR AN END USER .............................................. 74
  5.15.1 Request URI variables ................................................................. 75
  5.15.2 Response Codes ........................................................................ 75
    5.15.2.1 HTTP Response Codes ................................................... 75
    5.15.2.2 Exception fault codes ....................................................... 75
  5.15.3 GET.................................................................................................. 75
    5.15.3.1 Example: get all amount reservation transactions (Informative)
                    ................................................................................... 75
      5.15.3.1.1 Request ................................................................. 75
      5.15.3.1.2 Response .............................................................. 75
  5.15.4 PUT.................................................................................................. 77
  5.15.5 POST.............................................................................................. 77
    5.15.5.1 Example: create amount reservation (Informative) ......... 77
      5.15.5.1.1 Request ................................................................. 77
      5.15.5.1.2 Response .............................................................. 78
  5.15.6 DELETE............................................................................................ 78

5.16 RESOURCE: INDIVIDUAL AMOUNT RESERVATION TRANSACTION FOR AN END USER ....................................... 78
  5.16.1 Request URI variables ................................................................. 79
  5.16.2 Response Codes ........................................................................ 79
    5.16.2.1 HTTP Response Codes ................................................... 79
    5.16.2.2 Exception fault codes ....................................................... 79
  5.16.3 GET.................................................................................................. 79
    5.16.3.1 Example: get amount reservation (Informative) .......... 79
      5.16.3.1.1 Request ................................................................. 79
      5.16.3.1.2 Response .............................................................. 79
  5.16.4 PUT.................................................................................................. 80
  5.16.5 POST.............................................................................................. 80
    5.16.5.1 Example 1: charge amount for amount reservation (informative) 81
      5.16.5.1.1 Request ................................................................. 80
      5.16.5.1.2 Response .............................................................. 81
    5.16.5.2 Example 2: repeat a charge request with same referenceSequence for an amount reservation transaction
                    (Informative) 81
      5.16.5.2.1 Request ................................................................. 81
      5.16.5.2.2 Response .............................................................. 81
    5.16.5.3 Example 3: release amount reservation (Informative) .... 82
      5.16.5.3.1 Request ................................................................. 82
      5.16.5.3.2 Response .............................................................. 83
    5.16.5.4 Example 4: charge partial amount for amount reservation (Informative) 83
      5.16.5.4.1 Request ................................................................. 83
5.17 Resource: All Volume Reservation Transactions for an End User ................................................................. 88
5.17.1 Request URI variables .................................................................................................................................. 88
5.17.2 Response Codes .......................................................................................................................................... 88
5.17.2.1 HTTP Response Codes ........................................................................................................................... 88
5.17.2.2 Exception fault codes ............................................................................................................................... 88
5.17.3 GET .................................................................................................................................................................. 88
5.17.3.1 Example: get all volume reservation transactions (Informative) ............................................................ 88
5.17.3.1.1 Request ............................................................................................................................................... 88
5.17.3.1.2 Response ............................................................................................................................................ 88
5.17.4 PUT ................................................................................................................................................................ 90
5.17.5 POST ............................................................................................................................................................ 90
5.17.5.1 Example 1: create reserve volume (Informative) ..................................................................................... 90
5.17.5.1.1 Request ............................................................................................................................................... 90
5.17.5.1.2 Response ............................................................................................................................................ 90
5.17.5.2 Example 2: create reserve volume with invalid (non-existing) endUserID (Informative) ......................... 91
5.17.5.2.1 Request ............................................................................................................................................... 91
5.17.5.2.2 Response ............................................................................................................................................ 92
5.17.6 DELETE ..................................................................................................................................................... 92
5.18 Resource: Individual Volume Reservation Transaction for an End User ......................................................... 92
5.18.1 Request URI variables .................................................................................................................................. 92
5.18.2 Response Codes .......................................................................................................................................... 93
5.18.2.1 HTTP Response Codes ........................................................................................................................... 93
5.18.2.2 Exception fault codes ............................................................................................................................... 93
5.18.3 GET .................................................................................................................................................................. 93
5.18.3.1 Example: get volume reservation (Informative) ....................................................................................... 93
5.18.3.1.1 Request ............................................................................................................................................... 93
5.18.3.1.2 Response ............................................................................................................................................ 93
5.18.4 PUT ................................................................................................................................................................ 94
5.18.5 POST ............................................................................................................................................................ 94
5.18.5.1 Example: charge volume (Informative) ................................................................................................. 94
5.18.5.1.1 Request ............................................................................................................................................... 94
5.18.5.1.2 Response ............................................................................................................................................ 94
5.18.6 DELETE ..................................................................................................................................................... 95
5.19 Resource: Individual Amount for Volume Reservation Transaction for an End User ........................................ 95
5.19.1 Request URI variables .................................................................................................................................. 95
5.19.2 Response Codes .......................................................................................................................................... 95
5.19.2.1 HTTP Response Codes ........................................................................................................................... 95
5.19.2.2 Exception fault codes ............................................................................................................................... 96
5.19.3 GET .................................................................................................................................................................. 96
5.19.3.1 Example: get amount for volume reservation (Informative) ................................................................. 96
5.19.3.1.1 Request ............................................................................................................................................... 96
5.19.3.1.2 Response ............................................................................................................................................ 96
5.19.4 PUT ................................................................................................................................................................ 96
5.19.5 POST ............................................................................................................................................................ 96
5.19.6 DELETE ..................................................................................................................................................... 96
5.20 Resource: Amount Converted from Given Volume ............................................................................................ 96
5.20.1 Request URI variables .................................................................................................................................. 97
5.20.2 Response Codes .......................................................................................................................................... 97
5.20.2.1 HTTP Response Codes ........................................................................................................................... 97
## 6. FAULT DEFINITIONS

### 6.1 SERVICE EXCEPTIONS

### 6.2 POLICY EXCEPTIONS

### 6.2.1 POLICY EXCEPTIONS

- POL0252: Refund request failed
- POL0253: Payment operation refused by user

### APPENDIX A. CHANGE HISTORY (INFORMATIVE)

#### A.1 APPROVED VERSION HISTORY

#### A.2 DRAFT VERSION 1.0 HISTORY

### APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)

#### B.1 SCR FOR PARLAYREST.PAYMENT SERVER

- B.1.1 SCR for ParlayREST.Payment.Transactions Server
- B.1.2 SCR for ParlayREST.Payment.AmountCharge.Transactions Server
- B.1.3 SCR for ParlayREST.Payment.SplitAmount.Transactions Server
- B.1.4 SCR for ParlayREST.Payment.Individual.AmountCharge Server
- B.1.5 SCR for ParlayREST.Payment.Individual.SplitAmount Server
- B.1.6 SCR for ParlayREST.Payment.VolumeCharge.Transactions Server
- B.1.7 SCR for ParlayREST.Payment.Split.Volume.Transactions Server
- B.1.8 SCR for ParlayREST.Payment.Individual.VolumeCharge Server
- B.1.9 SCR for ParlayREST.Payment.Individual.Split.VolumeCharge Server
- B.1.10 SCR for ParlayREST.Payment.Individual.Amount.VolumeCharge Server
- B.1.12 SCR for ParlayREST.Payment.Amount.Reserve.Transactions Server
- B.1.16 SCR for ParlayREST.Payment.Individual.Amount.Volume Server
- B.1.17 SCR for ParlayREST.Payment.Amount Converted Server

### APPENDIX C. APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR SELECTED OPERATIONS

#### C.1 CHARGE AN AMOUNT

<table>
<thead>
<tr>
<th>Example (Informative)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>110</td>
</tr>
<tr>
<td>Response</td>
<td>111</td>
</tr>
</tbody>
</table>

#### C.2 REFUND AN AMOUNT

<table>
<thead>
<tr>
<th>Example (Informative)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>113</td>
</tr>
<tr>
<td>Response</td>
<td>114</td>
</tr>
</tbody>
</table>

#### C.3 RESERVE AN AMOUNT

<table>
<thead>
<tr>
<th>Example (Informative)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>116</td>
</tr>
<tr>
<td>Response</td>
<td>117</td>
</tr>
</tbody>
</table>

#### C.4 RESERVE AN ADDITIONAL AMOUNT

<table>
<thead>
<tr>
<th>Example (Informative)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>120</td>
</tr>
<tr>
<td>Response</td>
<td>121</td>
</tr>
</tbody>
</table>

#### C.5 CHARGE TO A RESERVATION

<table>
<thead>
<tr>
<th>Example (Informative)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>122</td>
</tr>
</tbody>
</table>

---

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.
## APPENDIX D. JSON EXAMPLES (INFORMATIVE) .................................................................................................................. 127

| D.1 | GET ALL TRANSACTIONS (SECTION 5.4.3.1) .................................................................................................................. 127 |
| D.2 | REQUEST WITH INVALID ENDUSERID (SECTION 5.4.3.2) ........................................................................................... 131 |
| D.3 | GET ALL AMOUNT TRANSACTIONS (SECTION 5.5.3.1) ..................................................................................................... 131 |
| D.4 | CREATE CHARGE AMOUNT (SECTION 5.5.5.1) ................................................................................................................ 133 |
| D.5 | CREATE REFUND AMOUNT (SECTION 5.5.5.2) ............................................................................................................... 134 |
| D.6 | CLIENT RETRIES POST WITH SAME CLIENTCORRELATOR (SECTION 5.5.5.3) ..................................................... 135 |
| D.7 | UNSUCCESSFUL CHARGE REQUEST BECAUSE OF DENIAL/REFUSAL BY BACK-END SYSTEM (SECTION 5.5.5.4) .... 135 |
| D.8 | GET ALL AMOUNT SPLIT TRANSACTIONS (SECTION 5.6.3.1) .................................................................................... 136 |
| D.9 | CREATE SPLIT CHARGE AMOUNT (SECTION 5.6.5.1) ................................................................................................. 138 |
| D.10 | GET AMOUNT CHARGE (SECTION 5.7.3.1) .................................................................................................................... 139 |
| D.11 | GET AMOUNT SPLIT CHARGE (SECTION 5.8.3.1) ......................................................................................................... 139 |
| D.12 | GET ALL VOLUME CHARGE AND REFUND TRANSACTIONS (SECTION 5.9.3.1) ..................................................... 140 |
| D.13 | CREATE CHARGE VOLUME, RETURNING A REPRESENTATION OF CREATED RESOURCE (SECTION 5.9.5.1) ............ 141 |
| D.14 | CREATE CHARGE VOLUME, RETURNING THE LOCATION OF CREATED RESOURCE (SECTION 5.9.5.2) .............. 142 |
| D.15 | CREATE REFUND VOLUME (SECTION 5.9.5.3) ............................................................................................................. 143 |
| D.16 | GET ALL VOLUME SPLIT CHARGE TRANSACTIONS (SECTION 5.10.3.1) ............................................................. 144 |
| D.17 | CREATE VOLUME SPLIT CHARGE (SECTION 5.10.5.1) ............................................................................................... 145 |
| D.18 | GET VOLUME CHARGE (SECTION 5.11.3.1) ................................................................................................................ 146 |
| D.19 | GET VOLUME SPLIT CHARGE (SECTION 5.12.3.1) ....................................................................................................... 147 |
| D.20 | GET AMOUNT FOR VOLUME CHARGE (SECTION 5.13.3.1) ....................................................................................... 148 |
| D.21 | GET AMOUNT FOR VOLUME SPLIT CHARGE (SECTION 5.14.3.1) ............................................................................ 148 |
| D.22 | GET ALL AMOUNT RESERVATION TRANSACTIONS (SECTION 5.15.3.1) ........................................................... 148 |
| D.23 | CREATE RESERVE AMOUNT (SECTION 5.15.5.1) .................................................................................................... 150 |
| D.24 | GET AMOUNT RESERVATION (SECTION 5.16.3.1) ...................................................................................................... 151 |
| D.25 | CHARGE AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.1) ............................................................... 152 |
| D.26 | REPEAT CHARGE REQUEST WITH SAME REFERENCESEQUENCE FOR AN AMOUNT RESERVATION (SECTION 5.16.5.2) 153 |
| D.27 | RELEASE AMOUNT RESERVATION (SECTION 5.16.5.3) .......................................................................................... 154 |
| D.28 | CHARGE PARTIAL AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.4) .................................................. 155 |
| D.29 | RELEASE REMAINING AMOUNT RESERVATION (SECTION 5.16.5.5) ................................................................. 156 |
| D.30 | RESERVE ADDITIONAL AMOUNT FOR AMOUNT RESERVATION (SECTION 5.16.5.6) ....................................... 156 |
| D.31 | UNSUCCESSFUL CHARGE AMOUNT FOR AMOUNT RESERVATION BECAUSE OF DENIAL/REFUSAL BY BACK-END SYSTEM (SECTION 5.16.5.7) ............................................................... 157 |
| D.32 | GET ALL VOLUME RESERVATION TRANSACTIONS (SECTION 5.17.3.1) .......................................................... 158 |
| D.33 | CREATE RESERVE VOLUME (SECTION 5.17.5.1) ...................................................................................................... 160 |
| D.34 | CREATE RESERVE VOLUME WITH INVALID ENDUSERID (SECTION 5.17.5.2) .................................................. 161 |
| D.35 | GET VOLUME RESERVATION (SECTION 5.18.3.1) ..................................................................................................... 161 |
| D.36 | CHARGE VOLUME (SECTION 5.18.5.1) ..................................................................................................................... 162 |
| D.37 | GET AMOUNT FOR VOLUME RESERVATION (SECTION 5.19.3.1) .......................................................................... 163 |
| D.38 | GET AMOUNT CONVERTED FROM VOLUME (SECTION 5.20.3.1) ......................................................................... 163 |

## APPENDIX E. PARLAY X OPERATIONS MAPPING (INFORMATIVE) .................................................................................. 165

## APPENDIX F. MAPPING OF PARLAYREST RESERVATION TRANSACTIONS TO THE CHARGING ENABLER (INFORMATIVE) ........................................................................................................ 166

| F.1 | CHARGING MECHANISM ........................................................................................................................................... 166 |
| F.2 | DETAILED MAPPING OF PARLAYREST TO CHARGING ENABLER ........................................................................ 166 |

### Figures

Figure 1 Resource structure defined by this specification .................................................................................................. 16
Figure 2 Amount charge and refund transaction.................................................................34
Figure 3 Amount split charge transaction...........................................................................35
Figure 4 Volume charge and refund transaction.................................................................36
Figure 5 Volume split charge transaction..........................................................................36
Figure 6 Amount reservation transaction..........................................................................37
Figure 7 Volume reservation transaction...........................................................................39
Figure 8 Amount converted from volume transaction.........................................................40
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


[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


3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACR</td>
<td>Anonymous Customer Reference</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>MSISDN</td>
<td>Mobile Subscriber ISDN Number</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>REST</td>
<td>REpresentational State Transfer</td>
</tr>
<tr>
<td>SCR</td>
<td>Static Conformance Requirements</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>TS</td>
<td>Technical Specification</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition</td>
</tr>
</tbody>
</table>
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’s account
- Reserving an amount or volume for an end user’s account
- Adding an amount or volume to an existing reservation
- Charging to a previously made reservation
- Releasing funds left in a previously made reservation
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). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

The remaining subsections in section 5 contain the detailed specification for each of the resources. Each such subsection defines the resource, the request URI variables that are common for all HTTP commands, the possible HTTP response codes, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 5 use XML as the format for the message body. Form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D. Appendix B provides the Static Conformance Requirements (SCR).

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

For requests and responses that have a body, the following applies: in the requests received, the server SHALL support JSON and XML encoding of the parameters in the body, and MAY support www-form-urlencoded parameters in the body. The Server SHALL return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST_TS_Common]. In notifications to the Client, the server SHALL use either XML or JSON encoding, depending on which format the client has specified in the related subscription.

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

5.1 Resources Summary

This section summarizes all the resources used by the Payment API.

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.
Figure 1 Resource structure defined by this specification

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

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/{api Version}/payment</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>All end user payment transactions</td>
<td>/{endUserId}/transactions</td>
<td>PaymentTransactionList</td>
<td>return all completed and pending payment transactions (amount, volume, amount reservation and volume reservation)</td>
</tr>
</tbody>
</table>

### Purpose: Amount charge, split charge, refund transactions

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

**Purpose: Volume charge, split charge, refund transactions**

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE</td>
</tr>
<tr>
<td>All volume charge and refund transactions for an end user</td>
<td>/{endUserId}//transactions/volume</td>
<td>PaymentTransactionList (used for GET)</td>
<td>return all volume transactions details for a given end user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VolumeTransaction (used for POST)</td>
<td>common:ResourceReference (optional alternative for POST response)</td>
</tr>
<tr>
<td>All volume split charge transactions for an end user</td>
<td>/{endUserId}//transactions/volumeSplit</td>
<td>PaymentTransactionList(used for GET)</td>
<td>return all volume split transactions details for a given end user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VolumeSplitTransaction (used for POST)</td>
<td>common:ResourceReference (optional alternative for POST response)</td>
</tr>
<tr>
<td>Individual volume charge or refund transaction for an end user</td>
<td>/{endUserId}//transactions/volume/{transactionId}</td>
<td>VolumeTransaction</td>
<td>return volume transaction details (charge, refund)</td>
</tr>
<tr>
<td>Individual volume split charge</td>
<td>/{endUserId}//transactions/volumeSplit/{transactionId}</td>
<td>VolumeSplitTransaction</td>
<td>return volume split transaction details</td>
</tr>
<tr>
<td>Resource</td>
<td>URL Base URL: http://{serverRoot}/{api Version}/payment</td>
<td>Data Structures</td>
<td>HTTP verbs</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------</td>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>transaction for an end user</td>
<td>/{endUserId}/transactions/volume/{transactionId}/paymentAmount</td>
<td>PaymentAmount</td>
<td>GET: charge)</td>
</tr>
<tr>
<td>Individual amount for volume charge or refund transaction for an end user</td>
<td>/{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount</td>
<td>PaymentAmount</td>
<td>GET: return transaction payment information based on provided volume; no PUT DELETE</td>
</tr>
<tr>
<td>Individual amount for volume split charge transaction for an end user</td>
<td>/{endUserId}/transactions/amountReservation/{transactionId}</td>
<td>AmountReservationTransaction</td>
<td>GET: return all amount reservation transactions details for a given end user; POST: create new reservation transaction for a given end user; no PUT DELETE</td>
</tr>
<tr>
<td>Purpose: Amount reservation transactions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>URL Base URL: http://{serverRoot}/{api Version}/payment</td>
<td>Data Structures</td>
<td>HTTP verbs</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------</td>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>All amount reservation transactions for an end user</td>
<td>/{endUserId}/transactions/amountReservation</td>
<td>PaymentTransactionList (used for GET) AmountReservationTransaction (used for POST) common:ResourceReference (optional alternative for POST response)</td>
<td>GET: return all amount reservation transactions details for a given end user; POST: create new reservation transaction for a given end user; no PUT DELETE</td>
</tr>
<tr>
<td>Individual amount reservation transaction for an end user</td>
<td>/{endUserId}/transactions/amountReservation/{transactionId}</td>
<td>AmountReservationTransaction</td>
<td>GET: return amount reservation transaction details (amount, charge); POST: Charge reserved amount, increase reservation amount, release reserved amount; no PUT DELETE</td>
</tr>
</tbody>
</table>
### Purpose: Volume reservation transactions

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>All volume reservation transactions for an end user</td>
<td>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation</td>
<td>PaymentTransactionList (used for GET) VolumeReservationTransaction (used for POST) common:ResourceReference (optional alternative for POST response)</td>
<td>GET: return transaction details for a given end user POST: create new volume reservation transaction for a given end user PUT: no DELETE: no</td>
</tr>
<tr>
<td>Individual volume reservation transaction for an end user</td>
<td>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</td>
<td>VolumeReservationTransaction</td>
<td>GET: return volume reservation transaction details (volume charge) POST: charge reserved volume, increase reserved volume or release reserved volume PUT: no DELETE: no</td>
</tr>
<tr>
<td>Individual amount for volume reservation transaction for an end user</td>
<td>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount</td>
<td>PaymentAmount</td>
<td>GET: return transaction payment information (calculates charge amount based on volume) POST: no DELETE: no</td>
</tr>
</tbody>
</table>

### Purpose: Amount converted from volume transaction

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount converted from given volume</td>
<td>http://{serverRoot}/{apiVersion}/payment/{endUserId}/convertedVolume/paymentAmount?volume={volume}&amp;unit={Minutes</td>
<td>Bytes</td>
<td>...}&amp;contract={contractId}&amp;service={serviceId}&amp;operation={operationId}</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amountTransaction</td>
<td>AmountTransaction</td>
<td>Yes</td>
<td>Collection of AmountTransaction</td>
</tr>
<tr>
<td>amountSplitTransaction</td>
<td>AmountSplitTransaction</td>
<td>Yes</td>
<td>Collection of AmountSplitTransaction</td>
</tr>
<tr>
<td>volumeTransaction</td>
<td>VolumeTransaction</td>
<td>Yes</td>
<td>Collection of VolumeTransaction</td>
</tr>
<tr>
<td>volumeSplitTransaction</td>
<td>VolumeSplitTransaction</td>
<td>Yes</td>
<td>Collection of VolumeSplitTransaction</td>
</tr>
<tr>
<td>volumeReservationTransaction</td>
<td>VolumeReservationTransaction</td>
<td>Yes</td>
<td>Collection of VolumeReservationTransaction</td>
</tr>
<tr>
<td>amountReservationTransaction</td>
<td>AmountReservationTransaction</td>
<td>Yes</td>
<td>Collection of AmountReservationTransaction</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Provided by the server and points to other resources that are in relationship with the current resource</td>
</tr>
</tbody>
</table>

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

5.2.2 Type: AmountTransaction

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The end user's account to be charged</td>
</tr>
<tr>
<td>paymentAmount</td>
<td>PaymentAmount</td>
<td>No</td>
<td>Information on the amount charge to be made</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>E.g. charged, refunded, etc</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.</td>
</tr>
<tr>
<td>serverReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).</td>
</tr>
<tr>
<td>originalServerReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The end user's account to be charged</td>
</tr>
<tr>
<td>paymentVolume</td>
<td>PaymentVolume</td>
<td>No</td>
<td>Information on the volume charge to be made</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>E.g. charged, refunded, etc</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to uniquely identify the request, e.g. in case of disputes. Used</td>
</tr>
</tbody>
</table>
for business logic, not for operational logic.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A unique reference to the request, provided by the server, and meaningful to the server’s backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request).</td>
</tr>
<tr>
<td>originalServerReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity.</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>The end user's account to be charged. It MUST be present in the initial reservation request.</td>
</tr>
<tr>
<td>paymentAmount</td>
<td>PaymentAmount</td>
<td>No</td>
<td>Information on the amount charge to be made</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>reserved, charged or released</td>
</tr>
<tr>
<td>referenceSequence</td>
<td>xsd:int</td>
<td>No</td>
<td>Sequential number generated by client application for every transaction state change</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>referenceCode</th>
<th>xsd:string</th>
<th>Yes</th>
<th>Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic.</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A unique reference to the request, provided by the server, and meaningful to the server’s backend system for correlation purposes.</td>
</tr>
<tr>
<td>originalServerReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>This may be used to reconcile a current transaction request with an original transaction.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self-referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Provided by the server and points to other resources that are in relationship with the current resource.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>The end user's account to be charged. It MUST be present in the initial reservation request.</td>
</tr>
<tr>
<td>paymentVolume</td>
<td>PaymentVolume</td>
<td>No</td>
<td>Information on the amount charge to be made</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>reserved, charged or released</td>
</tr>
</tbody>
</table>
| referenceSequence     | xsd:integer               | No       | Sequential number generated by client application for every transaction state change (e.g. reserve amount X (seq=1), reserve additional amount Y (seq=2), charge reserved amount (seq=3), etc). The client will increment reference sequence with every new request to the server. If request failed the client can repeat the request with the same sequence number. This allows the server to distinguish easily between new and repeated requests (e.g. ignore repeated requests, in the case they completed on the server.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
</tr>
<tr>
<td>serverReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
</tr>
<tr>
<td>originalServerReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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

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

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

5.2.9 Type: PaymentVolume

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>billingText</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to appear on the bill</td>
</tr>
<tr>
<td>volume</td>
<td>xsd:decimal</td>
<td>No</td>
<td>The volume to be charged</td>
</tr>
</tbody>
</table>
| 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:  
  - "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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onBehalfOf</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>purchaseCategoryCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>mandateId</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>serviceId</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>productId</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely indentify 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.</td>
</tr>
</tbody>
</table>
5.2.11 Type: EndUserShare

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The end user’s account to be charged</td>
</tr>
<tr>
<td>percent</td>
<td>xsd:integer</td>
<td>No</td>
<td>The percentage of this end user’s share. The sum of all shares must equal 100. The value of percentage should be positive.</td>
</tr>
</tbody>
</table>

5.2.12 Type: RatingParameter

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>xsd:string</td>
<td>No</td>
<td>Free string (e.g.: “unit”, “contract”, “service”, “operation”, etc)</td>
</tr>
<tr>
<td>value</td>
<td>xsd:string</td>
<td>No</td>
<td>Value of the rating parameter (e.g. “minutes” for “unit, “gold” for “contract”, “video” for “service”, etc)</td>
</tr>
</tbody>
</table>

5.2.13 Enumeration: TransactionOperationStatus

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

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

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

5.2.14 Values of the Link “rel” attribute

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

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

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

5.3 Sequence diagrams

5.3.1 Amount charge and refund transaction

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

The resource:

- To create an amount charge or refund transaction, create new resource under

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

![Figure 2 Amount charge and refund transaction](image-url)
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

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.

Figure 3 Amount split charge transaction

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

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.
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

  \[http://\{serverRoot\}/\{apiVersion\}/payment/\{endUserId\}/transactions/volumeSplit\]

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 transaction

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

The resources:

- To create an amount reservation transaction for an end user, create new resource under
  \[
  \text{http://}\{\text{serverRoot}\}/\{\text{apiVersion}\}/\text{payment}/\{\text{endUserId}\}/\text{transactions/amountReservation}
  \]

- To update an amount reservation transaction for an end user, use resource
  \[
  \text{http://}\{\text{serverRoot}\}/\{\text{apiVersion}\}/\text{payment}/\{\text{endUserId}\}/\text{transactions/amountReservation}/\{\text{transactionId}\}
  \]

---

**Figure 6 Amount reservation transaction**

1. POST ("reserved"): create new amount reservation
   
   Response ("reserved"): amount reservation transaction with transactionId

2. POST ("charged"): charge amount reservation with transactionId
   
   Response ("charged")

3. POST ("reserved"): reserve additional amount reservation with transactionId
   
   Response ("reserved")

4. POST ("charged"): charge amount reservation with transaction id
   
   Response ("charged")

5. POST ("released"): release amount reservation with transaction id
   
   Response ("released")
Outline of flow:

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

5.3.6 Volume reservation transaction

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

The resource:

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

- To update a volume reservation transaction for an end user, use resource
  
  http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}
Figure 7 Volume reservation transaction

Outline of flow:
1. Application creates a new volume reservation transaction for an end user. (POST)
2. Application charges a volume reservation with transactionId. (POST)
3. In case that add to the existing volume reservation, application reserves an additional volume reservation with transactionId. (POST)
4. Application charges a volume reservation with transactionId. (POST)
5. Application releases a volume reservation transaction for an end user. (POST)
5.3.7 Amount converted from volume transaction

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

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

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

\[
\text{http://[serverRoot]/[apiVersion]/payment/[endUserId]/convertedVolume/paymentAmount}
\]

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: \[\text{http://[serverRoot]/[apiVersion]/payment/[endUserId]/transactions}\]

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>

© 2010 Open Mobile Alliance Ltd. All Rights Reserved. Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document
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 29.199-6] and section 6 of the present document.

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

5.4.3.1 Example 1: get all transactions (Informative)

5.4.3.1.1 Request
GET ../{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 "Charged"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
  </amountTransaction>
</payment:paymentTransactionList>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>

<!-- COMPLETED AMOUNT REFUND TRANSACTION -->
<amountTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-789</serverReferenceCode>
  <clientCorrelator>54323</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
</amountSplitTransaction>

<!-- COMPLETED VOLUME CHARGE TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <!-- Additional volume transaction details -->
    </ratingParameter>
  </paymentVolume>
</volumeTransaction>
<volumeTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Refunded”</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>GHI-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54324</clientCorrelator>
  <resourceURL>http://{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 "Charged”</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>JKL-123</serverReferenceCode>
  <clientCorrelator>54325</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
</volumeSplitTransaction>

<!-- COMPLETED VOLUME REFUND TRANSACTION -->
<volumeTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Refunded”</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>GHI-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54324</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</volumeTransaction>
VOLUME RESERVATION TRANSACTION "RESERVED"

```xml
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>
```

VOLUME RESERVATION TRANSACTION "CHARGED"

```xml
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>MNO-123</serverReferenceCode>
  <clientCorrelator>66667</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>
```

COMPLETED AMOUNT RESERVATION TRANSACTION

```xml
<amountReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged"</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>25</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
</amountReservationTransaction>
```
5.4.3.2 Example 2: request with invalid (non-existing) 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/1.1 404 Not Found
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"/>
</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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user’s account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.
5.5.3 GET

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

5.5.3.1 Example: get all amount transactions (Informative)

5.5.3.1.1 Request

GET ../{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 "Charged"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST-012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>54321</clientCorrelator>
    <resourceURL>http://{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 "Charged"</description>
        <currency>USD</currency>
        <amount>10</amount>
        <code>TEST012345</code>
      </chargingInformation>
      <totalAmountCharged>10</totalAmountCharged>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>DEF-123</serverReferenceCode>
    <clientCorrelator>54321</clientCorrelator>
  </amountTransaction>
</payment:paymentTransactionList>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>

5.5.4 PUT

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

5.5.5 POST

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

5.5.5.1 Example 1: create charge amount (Informative)

5.5.5.1.1 Request

POST ../{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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST12345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>WXY-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54330</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</amountTransaction>
</payment:paymentTransactionList>
<amount>10</amount>
<code>TEST-012345</code>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

5.5.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 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">
<paymentAmount>
<chargingInformation>
<description>Test amount transaction "Charged"</description>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
</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">
<paymentAmount>
<chargingInformation>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
<description>Test amount transaction "Refunded"</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST-012345</code>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Refunded</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
<clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

5.5.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 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 "Refunded"</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST-012345</code>
</chargingInformation>
<totalAmountRefunded>10</totalAmountRefunded>
</paymentAmount>
<transactionOperationStatus>Refunded</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>XYZ-123</serverReferenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{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 [REST_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
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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

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

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

POST ../{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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>

5.5.5.4.2 Response

HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: 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>SVC0270</messageId>
    <text>Charging operation failed, the charge was not applied.</text>
  </serviceException>
</common:requestError>

5.5.6 DELETE

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

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

The resource used is:

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.6.3 GET

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

5.6.3.1 Example: get all amount split transactions (Informative)

5.6.3.1.1 Request

GET ../{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
<?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>
  </amountSplitTransaction>
</payment:paymentTransactionList>
```
5.6.4 PUT

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

5.6.5 POST

This operation is used to create a new transaction for an end user.
5.6.5.1 Example: create split charge amount

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"?>
<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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54431</clientCorrelator>
</payment:amountSplitTransaction>

5.6.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 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"?>
<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>
</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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>
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 29.199-6] and section 6 of the present document.

5.7.3 GET

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

5.7.3.1 Example: get amount charge (Informative)

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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}</resourceURL>
</payment:amountTransaction>
```
5.7.4 PUT

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

5.7.5 POST

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

5.7.6 DELETE

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

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

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.8.3 GET

This operation is used to return individual completed or pending amount split charge transaction information for an end user.
5.8.3.1 Example: get amount split charge (Informative)

5.8.3.1.1 Request

GET ../(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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}</resourceURL>
</payment:amountSplitTransaction>

5.8.4 PUT

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

5.8.5 POST

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

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

5.9 Resource: All volume charge and refund transactions for an end user

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.9.3 GET

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

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

5.9.3.1.1 Request

```
GET ../{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
```
5.9.4 PUT

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

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

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

5.9.5.1.1 Request

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


5.9.5.1.2 Response

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

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

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 "Charged" </billingText>
        <volume>10</volume>
        <ratingParameter>
            <name>unit</name>
            <value>minutes</value>
        </ratingParameter>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55551</clientCorrelator>
</payment:volumeTransaction>

5.9.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{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>

5.9.5.3 Example 3: create refund volume (Informative)

5.9.5.3.1 Request

POST ../{apiVersion}/payment/{endUserId}/transactions/volume HTTP/1.1
Accept: application/xml
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume transaction "Refunded"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.10.3 GET

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

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

5.10.3.1.1 Request

GET ../(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 "Charged"</billingText>
      <volume>10</volume>
      <ratingParameter>
        <name>unit</name>
        <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>55553</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
  </volumeSplitTransaction>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit</resourceURL>
</payment:paymentTransactionList>

5.10.4  PUT

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

5.10.5  POST

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

5.10.5.1  Example: create volume split charge  (Informative)

5.10.5.1.1  Request

POST ../(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 "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55553</clientCorrelator>
</payment:volumeSplitTransaction>

5.10.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 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 "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.11.3 GET

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

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 "Charged" </billingText>
        <volume>10</volume>
        <ratingParameter>
            <name>unit</name>
            <value>minutes</value>
        </ratingParameter>
        <totalVolumeCharged>10</totalVolumeCharged>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceCode>REF-12345</referenceCode>
    <serverReferenceCode>ABC-123</serverReferenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}</resourceURL>
</payment:volumeTransaction>

5.11.4 PUT

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

5.11.5 POST

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

5.11.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].
5.12 Resource: Individual volume split charge transaction for an end user

The resource used is:

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.12.3 GET

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

5.12.3.1 Example: get volume split charge (Informative)

5.12.3.1.1 Request

GET ../(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 "Charged"</billingText>
      <volume>10</volume>
      <ratingParameter>
         <name>unit</name>
         <value>minutes</value>
      </ratingParameter>
      <totalVolumeCharged>10</totalVolumeCharged>
   </paymentVolume>
   <transactionOperationStatus>Charged</transactionOperationStatus>
   <referenceCode>REF-12345</referenceCode>
   <serverReferenceCode>ABC-123</serverReferenceCode>
   <clientCorrelator>55556</clientCorrelator>
   <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}</resourceURL>
</payment:volumeSplitTransaction>

5.12.4 PUT

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

5.12.5 POST

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

5.12.6 DELETE

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

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

The resource used is:

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

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.13.3 GET

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

5.13.3.1 Example: get amount for volume charge  (Informative)

5.13.3.1.1 Request

GET /{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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.
5.14.3 GET

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

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

5.14.3.1.1 Request

GET ../{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 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 transactions for an end user.
5.15.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.15.3 GET

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

5.15.3.1 Example: get all amount reservation transactions (Informative)

5.15.3.1.1 Request

```
GET ../{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 "Charged"</description>
</chargingInformation>
</paymentAmount>
</amountReservationTransaction>
</payment:paymentTransactionList>
```
<currency>USD</currency>
<amount>15</amount>
<code>TEST012345</code>
</chargingInformation>
<totalAmountCharged>25</totalAmountCharged>
<amountReserved>0</amountReserved>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>55555</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</amountReservationTransaction>
<!-- AMOUNT RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL RESERVATION AMOUNT APPLIED SECOND TIME -->
5.15.4 PUT
Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].

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

5.15.5.1 Example: create amount reservation  (Informative)

5.15.5.1.1 Request

POST ../{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 "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>25</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</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 "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://{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 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 transaction for an end user.
5.16.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user’s account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.16.3 GET

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

5.16.3.1 Example: get amount reservation (Informative)

5.16.3.1.1 Request

GET ../{(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 "Reserved"</description>
    </chargingInformation>
  </paymentAmount>
</payment:amountReservationTransaction>
5.16.4 PUT

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

5.16.5 POST

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

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

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

5.16.5.1.1 Request

```
POST ../{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 "Charged" </description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>
```
5.16.5.2.2 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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55555</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

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

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

5.16.5.2.1 Request

POST ../{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 "Charged"</description>
    </chargingInformation>
  </paymentAmount>
</payment:amountReservationTransaction>
5.16.5.2.2 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 "Charged"</description>
            <currency>USD</currency>
            <amount>10</amount>
            <code>TEST012345</code>
        </chargingInformation>
        <totalAmountCharged>10</totalAmountCharged>
        <amountReserved>0</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Charged</transactionOperationStatus>
    <referenceSequence>2</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55555</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

5.16.5.3 Example 3: release amount reservation (Informative)

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

5.16.5.3.1 Request

POST ../{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 -->
5.16.5.3.2 Response

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

<?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 "Released"</description>  
<code>TEST012345</code>  
</chargingInformation>  
</paymentAmount>  
<transactionOperationStatus>Released</transactionOperationStatus>  
<referenceSequence>2</referenceSequence>  
</payment:amountReservationTransaction>

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

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

5.16.5.4.1 Request

POST ../{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 "Charged"</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>5</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

5.16.5.4.2 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 "Charged"</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>5</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55557</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>

5.16.5.5 Example 5: release remaining amount reservation (Informative)

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

POST ../{(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 "Released"</description>
      <code>TEST012345</code>
    </chargingInformation>
    <transactionOperationStatus>Released</transactionOperationStatus>
    <referenceSequence>3</referenceSequence>
  </paymentAmount>
</payment:amountReservationTransaction>

5.16.5.5.2 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"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Released"</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>0</amountReserved>
    <transactionOperationStatus>Released</transactionOperationStatus>
    <referenceSequence>3</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55558</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
  </paymentAmount>
</payment:amountReservationTransaction>
5.16.5.6  Example 6: reserve additional amount for amount reservation (Informative)

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

5.16.5.6.1  Request

POST ../{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 "Reserved"</description>
      <currency>USD</currency>
      <amount>5</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
</payment:amountReservationTransaction>

5.16.5.6.2  Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 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 "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
    </chargingInformation>
  </paymentAmount>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>15</amountReserved>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <clientCorrelator>55559</clientCorrelator>
</payment:amountReservationTransaction>
5.16.5.7 Example 7: unsuccessful charge amount for amount reservation because of denial/refusal by back-end system

(Informative)

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

5.16.5.7.1 Request

POST ../{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 "Charged"</description>
<currency>USD</currency>
<amount>10</amount>
<code>TEST012345</code>
</chargingInformation>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
</payment:amountReservationTransaction>

5.16.5.7.2 Response

HTTP/1.1 400 Bad Request
Content-Type: application/xml
Content-Length: 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="AmountReservationTransaction" href="http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"/>
<serviceException>
<messageId>SVC0270</messageId>
<text>Charging operation failed, the charge was not applied.</text>
</serviceException>
</common:requestError>

5.16.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC 2616].
5.17 Resource: All volume reservation transactions for an end user

The resource used is:

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

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

5.17.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.17.3 GET

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

5.17.3.1 Example: get all volume reservation transactions (Informative)

5.17.3.1.1 Request

GET ../{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 "RESERVED" -->
<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>

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

<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>15</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66667</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</volumeReservationTransaction>

<!-- VOLUME RESERVATION TRANSACTION "CHARGED" -->

<volumeReservationTransaction>
  <endUserId>tel:+16309700001</endUserId>
  <paymentVolume>
    <billingText>Test volume reservation transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
</volumeReservationTransaction>
5.17.4 PUT

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

5.17.5 POST

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

5.17.5.1 Example 1: create reserve volume (Informative)

5.17.5.1.1 Request

```
POST ../{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 "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>66666</clientCorrelator>
</payment:volumeReservationTransaction>
```

5.17.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
```
5.17.5.2 Example 2: create reserve volume with invalid (non-existing) endUserID

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 -->
<endUserId>tel:+16309700000</endUserId>
<paymentVolume>
  <billingText>Test volume reservation transaction "Reserved]}</billingText>
  <volume>10</volume>
  <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>0</totalVolumeCharged>
  <volumeReserved>10</volumeReserved>
</paymentVolume>
<transactionOperationStatus>Reserved</transactionOperationStatus>
<referenceSequence>1</referenceSequence>
<clientCorrelator>66667</clientCorrelator>
<resourceURL>http://(serverRoot)}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</payment:volumeReservationTransaction>
5.17.5.2.2 Response

HTTP/1.1 404 Not Found
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. The address %1 does not exist.</text>
    <variables>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 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 transaction for an end user.

5.18.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiKey</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document
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 29.199-6] and section 6 of the present document.

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

5.18.3.1 Example: get volume reservation (Informative)

5.18.3.1.1 Request
GET ../{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 "Reserved"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>0</totalVolumeCharged>
    <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</payment:volumeReservationTransaction>
5.18.4 PUT

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

5.18.5 POST

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

5.18.5.1 Example: charge volume (Informative)

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

5.18.5.1.1 Request

```plaintext
POST ../{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 "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:volumeReservationTransaction>
```

5.18.5.1.2 Response

```plaintext
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 "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
    </ratingParameter>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
</payment:volumeReservationTransaction>
```
<name>unit</name>
  <value>minutes</value>
</ratingParameter>
<totalVolumeCharged>10</totalVolumeCharged>
<volumeReserved>0</volumeReserved>
</paymentVolume>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceSequence>2</referenceSequence>
<referenceCode>REF-12345</referenceCode>
<clientCorrelator>66666</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}</resourceURL>
</payment:volumeReservationTransaction>

5.18.6 DELETE

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

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

The resource used is:
http://{serverRoot}/apiVersion/payment/{endUserId}/transactions/volumeReservation/{transactionId}/paymentAmount

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user’s account (xsd:anyURI)</td>
</tr>
<tr>
<td>transactionId</td>
<td>unique transaction identifier</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.19.3 GET

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

5.19.3.1 Example: get amount for volume reservation  (Informative)

5.19.3.1.1 Request

```
GET ../{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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>endUserId</td>
<td>the end user's account (xsd:anyURI)</td>
</tr>
</tbody>
</table>

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 29.199-6] and section 6 of the present document.

5.20.3 GET

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

Request URL parameters are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td>xsd:decimal</td>
<td>No</td>
<td>The given volume to be converted</td>
</tr>
<tr>
<td>unit</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Parameter to use when performing rating (e.g. &quot;minutes&quot;)</td>
</tr>
<tr>
<td>contract</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Parameter to use when performing rating (e.g. number of a contract that may govern the use)</td>
</tr>
<tr>
<td>service</td>
<td>xsd: string</td>
<td>Yes</td>
<td>Parameter to use when performing rating (e.g. VideoService&quot;)</td>
</tr>
<tr>
<td>operation</td>
<td>xsd: string</td>
<td>Yes</td>
<td>Parameter to use when performing rating (e.g. &quot;streamVideo&quot;)</td>
</tr>
</tbody>
</table>

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

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

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

6.1 Service Exceptions

None defined.

6.2 Policy Exceptions

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

6.2.1 POL0252: Refund request failed

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>POL0252</td>
</tr>
<tr>
<td>Text</td>
<td>Refund request failed: %1.</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 - textual description of failure. Value may be &quot;OriginalServerReferenceCode is required in refund request&quot;, &quot;Refund request amount exceeds original charge amount (charge amount included here)&quot;, &quot;Refunds not supported by implementation&quot;.</td>
</tr>
</tbody>
</table>

6.2.2 POL0253: Payment operation refused by user

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>POL0253</td>
</tr>
<tr>
<td>Text</td>
<td>Payment operation refused by user. %1</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 – additional information about the refusal</td>
</tr>
</tbody>
</table>
Appendix A. Change History

### A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>n/a</td>
<td>No prior version – or – No previous version within OMA</td>
</tr>
</tbody>
</table>

### A.2 Draft Version 1.0 History

<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>02 Dec 2009</td>
<td>5.2, 5.6, 5.10, 5.12, 5.14</td>
<td>Added OMA-ARC-REST-2009-0105R01-CR_Adding_Examples_ReserveAdditionalAmount_to_Payment_TS Added OMA-ARC-REST-2009-0110R03-CR_DataStructures_SplitRefund_to_Payment_TS.doc</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>07 Dec 2009</td>
<td>5.2</td>
<td>Added from OMA-ARC-REST-2009-0077-CR_Payment_API_Optionality.doc and added OMA-ARC-REST-2009-0113R01-CR_Outline_for_request_response_examples</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>15 Dec 2009</td>
<td>all</td>
<td>Added missing 151, and 129</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>16 Dec 2009</td>
<td>All</td>
<td>Editorial fixes: History table Styles as per template</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>26 Jan 2010</td>
<td>All</td>
<td>CONRR editorial comments applied, D0010, D0015</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>02 Feb 2010</td>
<td>All</td>
<td>Update OMA-ARC-REST-2010-0021</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>03 Feb 2010</td>
<td>all</td>
<td>Applied G001, G002, G004</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>04 Feb 2010</td>
<td>All</td>
<td>Added OMA-ARC-REST-2009-177R01</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>05 Feb 2010</td>
<td>All</td>
<td>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</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>18 Feb 2010</td>
<td>All</td>
<td>D006 (CR 54R01), D018 (CR 57R01), D019, (CR 65R01), D024 (CR 51R02), D043, editors</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST-Payment-V1_0</td>
<td>25 Feb 2010</td>
<td>All</td>
<td>82R02.</td>
</tr>
<tr>
<td>Document Identifier</td>
<td>Date</td>
<td>Sections</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>05 Mar 2010</td>
<td>All</td>
<td>CR 91, CR 99R01, CR 111R01</td>
</tr>
<tr>
<td></td>
<td>07 Mar 2010</td>
<td>All</td>
<td>Editorial (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))</td>
</tr>
<tr>
<td></td>
<td>08 Mar 2010</td>
<td>All</td>
<td>Editorial (table splits). Also added details on CRs applied in conjunction with CONRR comments (see above rows)</td>
</tr>
<tr>
<td></td>
<td>10 Mar 2010</td>
<td>Appendix D 5, App C</td>
<td>JSON examples added New style (‘listing’) applied to examples</td>
</tr>
<tr>
<td></td>
<td>11 Mar 2010</td>
<td>All</td>
<td>Applied decisions of Mar 10 walk-thru CC. Exception: removal of the last resource in the resource summary table (and related changes). Needs revisiting, because that was equivalent to the PX GetAmount.</td>
</tr>
<tr>
<td></td>
<td>16 Mar 2010</td>
<td>5.1, 5.3.7, 5.2, 5.20.3, Appendix C</td>
<td>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;</td>
</tr>
<tr>
<td></td>
<td>22 Mar 2010</td>
<td>5, 5.3</td>
<td>Implemented CR 134R01, editorials</td>
</tr>
<tr>
<td></td>
<td>27 Mar 2010</td>
<td>As per CRs</td>
<td>Implemented CR 144, 154R02 (using 137R02 agreed “style” for describing resource &amp; URL parameters), 158R01, 160, editorials</td>
</tr>
<tr>
<td>Candidate Version:</td>
<td>27 Apr 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP: OMA-TP-2010-0186- INP_ParlayREST_V1_0_ERP_for_Candidate_Approval</td>
</tr>
<tr>
<td></td>
<td>16 Jun 2010</td>
<td>55.1 5.2.13 5.3 D</td>
<td>Implemented agreed CRs: OMA-ARC-REST-2010-0187R04-CR_TransactionStatus_in_Payment_TS OMA-ARC-REST-2010-0223R01-CR_Payment_transition_states_for_reserve_and_release.doc OMA-ARC-REST-2010-0255R01-CR_Remove_PUT_on_amount_or_volume_charge_or_refund Editorial fix: History table styles</td>
</tr>
<tr>
<td></td>
<td>21 Jun 2010</td>
<td>Various</td>
<td>Fixed slanted quotes in the XML examples</td>
</tr>
<tr>
<td></td>
<td>22 Jun 2010</td>
<td>5.2.13</td>
<td>Added missing picture &amp; text from 0187R04</td>
</tr>
<tr>
<td>Document Identifier</td>
<td>Date</td>
<td>Sections</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>15 Jul 2010</td>
<td>Various</td>
<td>Implemented agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0292R01-CR_POST_instead_of_PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0337R01-CR_Transaction_Operation_Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0299R01-CR_XML_Validation_fix_Payment_TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0332-CR_Error_404_TS_Payment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0333R01-CR_Revision_of_CR293_Policy_exception_refund_not_supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0334R01-CR_Appendix_Reservation_Charging_CreditControl</td>
</tr>
<tr>
<td></td>
<td>21 Jul 2010</td>
<td>Various</td>
<td>Implemented agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0363-CR_Fix_form_urlencoded_example_errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0361R01-CR_Fix_SCR_errors_TS_Payment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0362-CR_Fixing_wrong_optionality_description_Appendix_C</td>
</tr>
<tr>
<td></td>
<td>24 Aug 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-TP-2010-0359-INP_ParlyREST_V1_0_ERP_for_Candidate_reapproval</td>
</tr>
<tr>
<td></td>
<td>24 Sep 2010</td>
<td>Various</td>
<td>Implemented agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0429R01-CR_resourceReference_in_resource_table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0453R01-CR_Fix_JSON_References_TS_Payment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0469-CR_Payment_add_resourceURL_description</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0473-CR_Payment_removing_PX_artifacts</td>
</tr>
<tr>
<td></td>
<td>04 Oct 2010</td>
<td>5 App C</td>
<td>Implemented agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0549-CR_Closing_Action_A67_TS_Payment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0522R02-CR_amount_in_Payment</td>
</tr>
<tr>
<td></td>
<td>08 Oct 2010</td>
<td>Various</td>
<td>Implemented agreed CR:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0563-CR_Payment_release_fixes Fixed [OMADICT]</td>
</tr>
<tr>
<td></td>
<td>14 Oct 2010</td>
<td>Various</td>
<td>Implemented agreed CR:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0566-CR_Payment_fix_3GPP_reference</td>
</tr>
<tr>
<td></td>
<td>23 Nov 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-TP-2010-0463R01-INP_ParlayREST_V1_0_ERP_for_Candidate_reapproval</td>
</tr>
</tbody>
</table>
Appendix B.  Static Conformance Requirements

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

B.1 SCR for ParlayREST.Payment Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-SUPPORT-S-001-M</td>
<td>Support for the Payment REST Enabler</td>
<td>5</td>
</tr>
<tr>
<td>PARLAYREST-PAY-SUPPORT-S-002-M</td>
<td>Support for the XML request &amp; response format</td>
<td>5</td>
</tr>
<tr>
<td>PARLAYREST-PAY-SUPPORT-S-003-M</td>
<td>Support for the JSON request &amp; response format</td>
<td>5</td>
</tr>
<tr>
<td>PARLAYREST-PAY-SUPPORT-S-004-O</td>
<td>Support for the application/form-urlencoded format</td>
<td>Appendix C</td>
</tr>
</tbody>
</table>

B.1.1 SCR for ParlayREST.Payment.Transactions Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-TRANS-S-001-O</td>
<td>Support for access to all completed and pending transactions</td>
<td>5.4 PARLAYREST-PAY-TRANS-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-TRANS-S-002-O</td>
<td>Obtain all completed and pending transactions - GET</td>
<td>5.4.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-001-M</td>
<td>Support for access to all amount charge and refund transactions for an end-user</td>
<td>5.5</td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-002-O</td>
<td>Obtain all amount charge and refund transactions for an end-user - GET</td>
<td>5.5.3</td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-003-M</td>
<td>Create a charge amount transaction for an end-user – POST (XML or JSON)</td>
<td>5.5.5</td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-004-O</td>
<td>Create a charge amount transaction for an end-user – POST (www-form-urlencoded)</td>
<td>C.1</td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-005-M</td>
<td>Create a refund amount transaction for an end-user - POST (XML or JSON)</td>
<td>5.5.5</td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-TRANS-S-006-O</td>
<td>Create a refund amount transaction for an end-user – POST (www-form-urlencoded)</td>
<td>C.2</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-S-001-O</td>
<td>Support access to individual amount charge or refund for an end-user</td>
<td>5.7</td>
<td>PARLAYREST-PAY-IND-AMNT-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-S-002-O</td>
<td>Retrieve individual completed or pending amount transaction - GET</td>
<td>5.7.3</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-SPLIT-S-001-O</td>
<td>Support access to individual split amount charge for an end-user</td>
<td>5.8</td>
<td>PARLAYREST-PAY-IND-SPLIT-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-SPLIT-S-002-O</td>
<td>Retrieve individual completed or pending split amount transaction - GET</td>
<td>5.8.3</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

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

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-VOL-S-001-O</td>
<td>Support access to individual volume charge or refund transaction for an end-user</td>
<td>5.11</td>
<td>PARLAYREST-PAY-IND-VOL-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-VOL-S-002-O</td>
<td>Retrieve individual completed or pending transaction - GET</td>
<td>5.11.3</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-SPLIT-VOL-S-001-O</td>
<td>Support access to individual split volume charge for an end-user</td>
<td>5.12</td>
<td>PARLAYREST-PAY-IND-SPLIT-VOL-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-SPLIT-VOL-S-002-O</td>
<td>Retrieve individual completed or pending split volume charge transaction - GET</td>
<td>5.12.3</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-VOL-S-001-O</td>
<td>Support access to individual payment information based on</td>
<td>5.13</td>
<td>PARLAYREST-PAY-IND-AMNT-VOL-S-002-O</td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
<td>Requirement</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-VOL-S-002-O</td>
<td>Retrieve individual payment information based on volume transaction - GET</td>
<td>5.13.4</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-001-O</td>
<td>Support access to individual payment information based on volume split transaction</td>
<td>5.14</td>
<td>PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-IND-AMNT-SPLIT-VOL-S-002-O</td>
<td>Retrieve individual payment information based on volume split transaction - GET</td>
<td>5.14.4</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-AMNT-RES-TRANS-S-001-M</td>
<td>Support for access to all completed and pending amount reservation transactions</td>
<td>5.15</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-RES-TRANS-S-002-O</td>
<td>Obtain all transactions for an end-user - GET</td>
<td>5.15.4</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-RES-TRANS-S-003-M</td>
<td>Create a new amount reservation transaction for an end-user – POST (XML or JSON)</td>
<td>5.15.6</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-PAY-AMNT-RES-TRANS-S-004-O</td>
<td>Create a new amount reservation transaction for an end-user – POST (www-form-urlencoded)</td>
<td>C.3</td>
<td></td>
</tr>
</tbody>
</table>


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


<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-VOL-RES-TRANS-S-001-O</td>
<td>Support for access to all volume reservation transactions</td>
<td>5.17</td>
<td>PARLAYREST-PAY-VOL-RES-TRANS-S-003-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-VOL-RES-TRANS-S-002-O</td>
<td>Obtain all transactions for an end-user - GET</td>
<td>5.17.4</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-PAY-VOL-RES-TRANS-S-003-O</td>
<td>Create a new volume reservation transaction for an end-user - POST</td>
<td>5.17.6</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-PAY-VOL-IND-RES-TRANS-S-001-O</td>
<td>Support for access to individual volume reservation transaction</td>
<td>5.18</td>
<td>PARLAYREST-PAY-VOL-IND-RES-TRANS-S-003-O</td>
</tr>
<tr>
<td>PARLAYREST-PAY-VOL-IND-RES-TRANS-S-002-O</td>
<td>Obtain individual transaction for an end-user - GET</td>
<td>5.18.4</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-PAY-VOL-IND-RES-TRANS-S-003-O</td>
<td>Update a volume reservation transaction for an end-user - POST</td>
<td>5.18.6</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
</table>
### 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 |
--- | --- | --- | ---
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’s account
- Reserving a charge for an end user’s account
- Adding/subtracting a charge to/from an existing reservation
- Charging to a previously made reservation
- Releasing funds left in a previously made reservation

C.1 Charge an Amount

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

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>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.</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>xsd:string</td>
<td>No</td>
<td>charged (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>currency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The currency of the amount</td>
</tr>
<tr>
<td>amount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The amount to be charged. The amount to be charged appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a...</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onBehalfOf</td>
<td>xsd:string</td>
<td>Yes</td>
<td>String parameter to allow aggregator or acquiring partners to specify who the payment is really by.</td>
</tr>
<tr>
<td>purchaseCategoryCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A category defining the type of service, product or media being purchased.</td>
</tr>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.</td>
</tr>
<tr>
<td>mandateID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID representing the subscription service or consent approval for which this charge applies. How the consent is established is out of scope.</td>
</tr>
<tr>
<td>serviceID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID of the partner/merchant service being purchased.</td>
</tr>
<tr>
<td>productID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely indentify the product being purchased.</td>
</tr>
</tbody>
</table>

**C.1.1 Example**

**C.1.1.1 Request**

POST ../(api/Version)/paymen/(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&
transactionOperationStatus=Charged&
description= Test%20amount%20transaction%20%22Charged%22&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
C.1.1.2 Response

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

```xml
<?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 "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{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.

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Unique identifier for the end user’s account</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>refunded(see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)</td>
</tr>
<tr>
<td>Description</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>currency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The currency of the amount</td>
</tr>
<tr>
<td>amount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The amount to be refunded. The amount to be refunded appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>originalServerReferenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>onBehalfOf</td>
<td>xsd:string</td>
<td>Yes</td>
<td>String parameter to allow</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>purchaseCategoryCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A category defining the type of service, product or media being purchased.</td>
</tr>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.</td>
</tr>
<tr>
<td>mandateID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID representing the subscription service or consent approval for which this charge applies.</td>
</tr>
<tr>
<td>serviceID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID of the partner/merchant service being purchased.</td>
</tr>
<tr>
<td>productId</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely indentify the product being purchased.</td>
</tr>
</tbody>
</table>

### 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&
transactionDateStatus=Refunded&
description= Test%20amount%20transaction%20%22Refunded%22&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
originalServerReferenceCode=ABC123&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
```
C.2.1.2 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"?>
<!DOCTYPE payment:amountTransaction SYSTEM http://_serverspecification/ proposed/urn:oma:xml:rest:payment:1.dtd>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:+16309700001</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Charged"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>DEF-123</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{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.

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Unique identifier for the end user’s account</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>reserved (see TransactionOperationStatus enumeration, section 5.2.13 for</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>currency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The currency of the amount</td>
</tr>
<tr>
<td>amount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The amount to be reserved. The amount to be reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceSequence</td>
<td>xsd:string</td>
<td>No</td>
<td>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).</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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</td>
</tr>
</tbody>
</table>
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.

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

### C.3.1 Example

#### C.3.1.1 Request

```plaintext
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
```
C.3.1.2 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 "Reserved"</description>
<currency>USD</currency>
<amount>10</amount>
</chargingInformation>
<totalAmountCharged>0</totalAmountCharged>
<amountReserved>10</amountReserved>
<chargingMetaData>
<onBehalfOf>Example Games Inc</onBehalfOf>
<purchaseCategoryCode>Game</purchaseCategoryCode>
<channel>WAP</channel>
<taxAmount>0</taxAmount>
</chargingMetaData>
</paymentAmount>
<transactionOperationStatus>Reserved</transactionOperationStatus>
<referenceSequence>1</referenceSequence>
<referenceCode>TEST-012345</referenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}</resourceURL>
</payment:amountReservationTransaction>
C.4 Reserve an Additional Amount

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

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Unique identifier for the end user’s account</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>e.g. charged, reserved, etc (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings)</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>currency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The currency of the amount</td>
</tr>
<tr>
<td>amount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The amount to be reserved. The amount to be reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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</td>
</tr>
</tbody>
</table>
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.

| 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.

<p>| purchaseCategoryCode | xsd:string | Yes | A category defining the type of service, product or media being purchased. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.</td>
</tr>
<tr>
<td>mandateID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID representing the subscription service or consent approval for which this charge applies.</td>
</tr>
<tr>
<td>serviceID</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>The ID of the partner/merchant service being purchased.</td>
</tr>
<tr>
<td>productID</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely indentify the product being purchased.</td>
</tr>
</tbody>
</table>

**C.4.1 Example (Informative)**

**C.4.1.1 Request**

```
POST ../{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

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Reserved"</description>
      <amount>5</amount>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <resourceURL>http://{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.

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Unique identifier for the end user’s account</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>charged (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>currency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The currency of the amount</td>
</tr>
<tr>
<td>amount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The amount to be charged. The amount to be charged appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>No</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>referenceSequence</td>
<td>xsd:int</td>
<td>No</td>
<td>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).</td>
</tr>
<tr>
<td>onBehalfOf</td>
<td>xsd:string</td>
<td>Yes</td>
<td>String parameter to allow aggregator or acquiring partners to specify who the payment is really by.</td>
</tr>
<tr>
<td>purchaseCategoryCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A category defining the type of service, product or media being purchased.</td>
</tr>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.</td>
</tr>
<tr>
<td>mandateID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID representing the subscription service or consent approval for which this charge applies.</td>
</tr>
<tr>
<td>serviceID</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>The ID of the partner/merchant service being purchased.</td>
</tr>
<tr>
<td>productID</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely identify the product being purchased.</td>
</tr>
</tbody>
</table>

**C.5.1 Example** *(Informative)*

**C.5.1.1 Request**

POST ../{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
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"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Charged"</description>
      <amount>5</amount>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
    <amountReserved>5</amountReserved>
    <chargingMetaData>
      <onBehalfOf>Example Games Inc</onBehalfOf>
      <purchaseCategoryCode>Game</purchaseCategoryCode>
      <channel>WAP</channel>
      <taxAmount>0</taxAmount>
    </chargingMetaData>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>3</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://{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.

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endUserId</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Unique identifier for the end user’s</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>transactionOperationStatus</td>
<td>TransactionOperationStatus</td>
<td>No</td>
<td>released (see TransactionOperationStatus enumeration, section 5.2.13 for allowed strings and description)</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>No</td>
<td>Information that appear on the bill</td>
</tr>
<tr>
<td>code</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code referencing a contract under which the charge is applied</td>
</tr>
<tr>
<td>referenceSequence</td>
<td>xsd:int</td>
<td>No</td>
<td>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).</td>
</tr>
<tr>
<td>referenceCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Textual information to uniquely identify the request, for example, in the case of disputes</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>onBehalfOf</td>
<td>xsd:string</td>
<td>Yes</td>
<td>String parameter to allow aggregator or acquiring partners to specify who the payment is really by.</td>
</tr>
<tr>
<td>purchaseCategoryCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A category defining the type of service, product or media being purchased.</td>
</tr>
<tr>
<td>channel</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP,</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>taxAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.</td>
</tr>
<tr>
<td>mandateID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID representing the subscription service or consent approval for which this charge applies.</td>
</tr>
<tr>
<td>serviceID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The ID of the partner/merchant service being purchased.</td>
</tr>
<tr>
<td>productID</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Combines with the service ID to uniquely identify the product being purchased.</td>
</tr>
</tbody>
</table>

### C.6.1 Example

#### C.6.1.1 Request

```
POST ../{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

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

#### C.6.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 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">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Released"</description>
    </chargingInformation>
    <totalAmountCharged>5</totalAmountCharged>
    <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Released</transactionOperationStatus>
</payment:amountReservationTransaction>
```
<referenceSequence>4</referenceSequence>
<resourceURL>http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation{transactionId}</resourceURL>
</payment:amountReservationTransaction>
Appendix D. JSON examples

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for Parlay REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC 4627].

The following examples show the request and response for various operations using a JSON binding. The examples follow the XML to JSON serialization rules in [REST_TS_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST_TS_Common].

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

D.1 Get all transactions (section 5.4.3.1)

Request:

```
GET ../{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": [
  {
   "clientCorrelator": "55555",
   "endUserId": "tel:+16309700001",
   "paymentAmount": {
    "amountReserved": "0",
    "chargingInformation": {
     "amount": "15",
     "code": "TEST012345",
     "currency": "USD",
     "description": "Test amount reservation transaction "Charged"
    },
    "totalAmountCharged": "25"
   },
   "referenceCode": "REF-12345",
   "referenceSequence": "2",
   "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
   "serverReferenceCode": "PQR-123",
   "transactionOperationStatus": "Charged"
  },
  {
   "clientCorrelator": "55556",
   "endUserId": "tel:+16309700001",
   "paymentAmount": {
    "amountReserved": "10",
    "chargingInformation": {
     "amount": "10",
    },
    "totalAmountCharged": "10"
   }
```
"code": "TEST012345",
"currency": "USD",
"description": "Test amount reservation transaction \"Reserved\"
},
"totalAmountCharged": "0"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionOperationStatus": "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 \"Charged\"
  },
  "totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountSplit/{transactionId}",
"serverReferenceCode": "ABC-789",
"transactionOperationStatus": "Charged"
},
"amountTransaction": [
  {
  "clientCorrelator": "54321",
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
  }
]
{"clientCorrelator": "54322",
"endUserId": "tel:+16309700001",
"originalServerReferenceCode": "ABC-123",
"paymentAmount": {
"chargingInformation": {
"amount": "10",
"code": "TEST012345",
"currency": "USD",
"description": "Test amount transaction \"Refunded\"
},
"totalAmountRefunded": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}",
"serverReferenceCode": "ABC-456",
"transactionOperationStatus": "Refunded",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions",
"volumeReservationTransaction": [
{"clientCorrelator": "66666",
"endUserId": "tel:+16309700001",
"paymentVolume": {
"billingText": "Test volume reservation transaction \"Reserved\"
},
"ratingParameter": {
"name": "unit",
"value": "minutes"
},
"totalVolumeCharged": "0",
"volume": "10",
"volumeReserved": "10"
},
"referenceCode": "REF-12345",
"referenceSequence": "1",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
"transactionOperationStatus": "Reserved"
},
{"clientCorrelator": "66667",
"endUserId": "tel:+16309700001",
"paymentVolume": {
"billingText": "Test volume reservation transaction \"Charged\"
},
"ratingParameter": {
"name": "unit",
"value": "minutes"
},
"totalVolumeCharged": "25",
"volume": "10",
"volumeReserved": "0"
},
"referenceCode": "REF-12345",
"referenceSequence": "4",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
"serverReferenceCode": "MNO-123"}
"transactionOperationStatus": "Charged"
],
"volumeSplitTransaction": {
  "clientCorrelator": "54325",
  "endUserShare": [
    {
      "endUserId": "tel:+16309700001",
      "percent": "20"
    },
    {
      "endUserId": "tel:+16309700002",
      "percent": "80"
    }
  ],
  "paymentVolume": {
    "billingText": "Test volume transaction \"Charged\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}"  
  "serverReferenceCode": "JKL-123",
  "transactionOperationStatus": "Charged"
},
"volumeTransaction": [
  {
    "clientCorrelator": "55551",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
    "serverReferenceCode": "DEF-123",
    "transactionOperationStatus": "Charged"
  },
  {
    "clientCorrelator": "54324",
    "endUserId": "tel:+16309700001",
    "originalServerReferenceCode": "DEF-123",
    "paymentVolume": {
      "billingText": "Test volume transaction \"Refunded\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "10",
      "volume": "10"
    },
    "referenceCode": "DEF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
    "serverReferenceCode": "DEF-123",
    "transactionOperationStatus": "Charged"
  }
]
D.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/1.1 404 Not Found
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": "Invalid input value. The address %1 does not exist.",
    "variables": "tel:+016309700000"
  }
}}
```
{"paymentTransactionList": {
  "amountTransaction": [
    {
      "clientCorrelator": "54321",
      "endUserId": "tel:+16309700001",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST-012345",
          "currency": "USD",
          "description": "Test amount transaction ("Charged")"
        },
        "totalAmountCharged": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
    },
    {
      "clientCorrelator": "54321",
      "endUserId": "tel:1234567890",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount transaction ("Charged")"
        },
        "totalAmountCharged": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
    },
    {
      "clientCorrelator": "54330",
      "endUserId": "tel:+16309700001",
      "originalServerReferenceCode": "DEF-123",
      "paymentAmount": {
        "chargingInformation": {
          "amount": "10",
          "code": "TEST012345",
          "currency": "USD",
          "description": "Test amount transaction ("Refunded")"
        },
        "totalAmountRefunded": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
    }
  ]
}}
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount"
}

## D.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 \"Charged\"
},
"referenceCode": "REF-12345",
"transactionOperationStatus": "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 \"Charged\"
},
"totalAmountCharged": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}
```
D.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",
    "originalServerReferenceCode": "ABC-123",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Refunded\""
        }
    },
    "referenceCode": "REF-12345",
    "transactionOperationStatus": "Refunded"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 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",
    "originalServerReferenceCode": "ABC-123",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Refunded\""
        },
        "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
}}
D.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 \"Charged\" "
    }
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "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 \"Charged\" "
    }
  },
  "totalAmountcharged": "10",
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/{transactionId}"
}}
```

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

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 \"Charged\" "  
  }},  
  "referenceCode": "REF-12345",  
  "transactionOperationStatus": "Charged"  
}}  

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": "SVC0270",  
  "text": "Charging operation failed, the charge was not applied."  
}}}

D.8 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 \"Charged\""
},
"totalAmountCharged": "20"
},
"referenceCode": "REF-12345",
"resourceURL": "http://(serverRoot)/(apiVersion)/payment/(endUserId)/transactions/amountSplit/(transactionId)",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "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 \"Charged\""
},
"totalAmountCharged": "10"
},
"referenceCode": "REF-12346",
"resourceURL": "http://(serverRoot)/(apiVersion)/payment/(endUserId)/transactions/amountSplit/(transactionId)",
"serverReferenceCode": "DEF-123",
"transactionOperationStatus": "Charged"
}
],
"resourceURL": "http://(serverRoot)/(apiVersion)/payment/(endUserId)/transactions/amountSplit"}
D.9 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:+16309700002",
      "percent": "70"
    }
  ],
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\""
    },
    "referenceCode": "REF-12345",
    "transactionOperationStatus": "Charged"
  }
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 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 \"Charged\""
    },
    "referenceCode": "REF-12345",
    "transactionOperationStatus": "Charged"
  }
}
D.10 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 \"Charged\"",
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amount/"{transactionId}"",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
    }
}}

D.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
D.12 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": [}
D.13 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 \"Charged\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
},

{"clientCorrelator": "55552",
  "endUserId": "tel:+16309700001",
  "originalServerReferenceCode": "ABC-123",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Refunded\" ",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeRefunded": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
  "serverReferenceCode": "DEF-123",
  "transactionOperationStatus": "Refunded"
}
}
```
Response:

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

{"volumeTransaction": {
    "clientCorrelator": "55551",
    "endUserId": "tel:+16309700001",
    "paymentVolume": {
        "billingText": "Test volume transaction "Charged"",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
}}

D.14 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 "Charged"",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
}}
D.15 Create refund volume (section 5.9.5.3)

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",
  "originalServerReferenceCode": "ABC-123",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Refunded\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Refunded"
}}
```

Response:

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

{"volumeTransaction": {
  "clientCorrelator": "55552",
  "endUserId": "tel:+16309700001",
  "originalServerReferenceCode": "ABC-123",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Refunded\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Refunded"
}}
```
"clientCorrelator": "55552",
"endUserId": "tel:+16309700001",
"originalServerReferenceCode": "ABC-123",
"paymentVolume": {
  "billingText": "Test volume transaction \"Refunded\"",
  "ratingParameter": {
    "name": "unit",
    "value": "minutes"
  }
},
"totalVolumeRefunded": "10",
"volume": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volume/{transactionId}",
"serverReferenceCode": "DEF-123",
"transactionOperationStatus": "Refunded"
}

D.16 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 \"Charged\"",
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      }
    }
  }
}
D.17 Create volume split charge (section 5.10.5.1.)

Request:

```json
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 \"Charged\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 201 Created
```
D.18 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",
"percent": "20"
},
{"endUserId": "tel:+16309700002",
"percent": "80"
},
"paymentVolume": {
"billingText": "Test volume transaction \"Charged\"",
"ratingParameter": {
"name": "unit",
"value": "minutes"
},
"volume": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}"
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"}}
D.19 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 \"Charged\" ",
"ratingParameter": {
"name": "unit",
"value": "minutes"
},
"totalVolumeCharged": "10",
"volume": "10"
},
"referenceCode": "REF-12345",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeSplit/{transactionId}",
"serverReferenceCode": "ABC-123",
"transactionOperationStatus": "Charged"
}}
D.20 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

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

D.21 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.22 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 \"Charged\" \n"
          },
          "totalAmountCharged": "25"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "2",
        "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
      },
      {
        "clientCorrelator": "55556",
        "endUserId": "tel:+16309700001",
        "paymentAmount": {
          "amountReserved": "10",
          "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Reserved\""
          },
          "totalAmountCharged": "0"
        },
        "referenceSequence": "1",
        "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
        "transactionOperationStatus": "Reserved"
      },
      {
        "clientCorrelator": "55557",
        "endUserId": "tel:+16309700001",
        "paymentAmount": {
          "amountReserved": "15",
          "chargingInformation": {
            "amount": "15",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Reserved\"
          },
          "totalAmountCharged": "0"
        },
        "referenceSequence": "3",
        "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
        "transactionOperationStatus": "Reserved"
      }
    ]
]
D.23 Create reserve amount (section 5.15.5.1)

Request:

```json
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": {
    "amountReserved": "25",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Reserved\"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "3",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"
}

"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation"
```
D.24 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": "TEST-012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Reserved\"
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"
  "transactionOperationStatus": "Reserved"
}
D.25 Charge amount for amount reservation (section 5.16.5.1)

Request:

POST ../{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": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {  
    "amount": "10",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Charged\""
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 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 \"Charged\"
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionOperationStatus": "Reserved"
}}
D.26 Repeat charge request with same referenceSequence for an amount reservation (section 5.16.5.2)

Request:

POST ../{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": {
    "endUserId": "tel:+16309700001",
    "paymentAmount": {"chargingInformation": {
        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \"Charged\""
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "transactionOperationStatus": "Charged"
}}

Response:

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

{"amountReservationTransaction": {
    "clientCorrelator": "55555",
    "endUserId": "tel:+16309700001",
    "paymentAmount": {"chargingInformation": {
        "amount": "10",
        "code": "TEST012345",
        "currency": "USD",
        "description": "Test amount reservation transaction \"Charged\""
    },
    "totalAmountCharged": "10"
},
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "transactionOperationStatus": "Charged"}
D.27 Release amount reservation (section 5.16.5.3)

Request:

POST ../{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": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {
    "code": "TEST012345",
    "description": "Test amount reservation transaction \"Released\"
  }},
  "referenceSequence": "2",
  "transactionOperationStatus": "Released"
}}

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 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 \"Released\"
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"
  "transactionOperationStatus": "Released"
}}
D.28 Charge partial amount for amount reservation (section 5.16.5.4)

Request:

POST ../{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": {
"endUserId": "tel:+16309700001",
"paymentAmount": {"chargingInformation": {
"amount": "5",
"code": "TEST012345",
"currency": "USD",
"description": "Test amount reservation transaction "Charged" "
}},
"referenceCode": "REF-12345",
"referenceSequence": "2",
"transactionOperationStatus": "Charged"
}}

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 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 "Charged" " 
},
"totalAmountCharged": "5"
},
"referenceCode": "REF-12345",
"referenceSequence": "2",
"resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
"transactionOperationStatus": "Charged"
}}
D.29 Release remaining amount reservation (section 5.16.5.5)

Request:

POST ../{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": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {
  "code": "TEST012345",
  "description": "Test amount reservation transaction \"Released\"
  }
  },
  "referenceSequence": "3",
  "transactionOperationStatus": "Released"
}}

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 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 \"Released\"
  }
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "3",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"
  },
  "transactionOperationStatus": "Released"
}}

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

Request:

POST ../{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": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {"chargingInformation": {
    "amount": "5",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction "Reserved"
  }},
  "referenceSequence": "2",
  "transactionOperationStatus": "Reserved"
}}

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 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 "Reserved"
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "2",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}",
  "transactionOperationStatus": "Reserved"
}}

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

Request:

POST ../{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": {
  "endUserId": "tel:+16309700001",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction "Charged""
    }
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "transactionOperationStatus": "Charged"
}}

Response:

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

{"requestError": {
  "link": {
    "href": "http://{serverRoot}/apiVersion}/payment/{endUserId}/transactions/amountReservation/{transactionId}"
  },
  "serviceException": {
    "messageId": "SVC0270",
    "text": "Charging operation failed, the charge was not applied."
  }
}}

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

Request:

GET ../{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 "Reserved"",
        "ratingParameter": {
          "name": "unit",
          "value": "minutes"
        },
        "totalVolumeCharged": "0",
        "volume": "10",
        "volumeReserved": "10"
      },
      "referenceSequence": "1",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
      "transactionOperationStatus": "Reserved"
    },
    {
      "clientCorrelator": "66667",
      "endUserId": "tel:+16309700001",
      "paymentVolume": {
        "billingText": "Test volume reservation transaction "Reserved"",
        "ratingParameter": {
          "name": "unit",
          "value": "minutes"
        },
        "totalVolumeCharged": "15",
        "volume": "10",
        "volumeReserved": "10"
      },
      "referenceCode": "REF-12345",
      "referenceSequence": "3",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
      "transactionOperationStatus": "Reserved"
    },
    {
      "clientCorrelator": "66668",
      "endUserId": "tel:+16309700001",
      "paymentVolume": {
        "billingText": "Test volume reservation transaction "Charged"
      },
      "ratingParameter": {
        "name": "unit",
        "value": "minutes"
      },
      "totalVolumeCharged": "25",
      "volume": "10",
      "volumeReserved": "0"
    },
    {
      "referenceCode": "REF-22345",
      "referenceSequence": "4",
      "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}",
      "serverReferenceCode": "ABC-123",
      "transactionOperationStatus": "Charged"
    }
  ]
}
D.33 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 \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "volume": "10"
  },
  "referenceSequence": "1",
  "transactionOperationStatus": "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": "66666",
  "endUserId": "tel:+16309700001",
  "paymentVolume": {
    "billingText": "Test volume reservation transaction \"Reserved\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeCharged": "0",
    "volume": "10",
    "volumeReserved": "10"
  },
  "referenceSequence": "1",
  "resourceURL": "http://{serverRoot}/{apiVersion}/payment/{endUserId}/transactions/volumeReservation/{transactionId}"}
D.34 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 \"Reserved\"
  }
},
"referenceSequence": "1",
"transactionOperationStatus": "Reserved"
}

Response:

HTTP/1.1 404 Not Found
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC0004",
  "text": "Invalid input value. The address %1 does not exist.",
  "variables": "tel:+016309700000"
}}}
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 \"Reserved\"",  
  "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}",  
"transactionOperationStatus": "Reserved"  
}}

D.36 Charge volume (section 5.18.5.1)

Request:

POST ../{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": {  
"endUserId": "tel:+16309700001",  
"paymentVolume": {  
  "billingText": "Test volume reservation transaction \"Charged\"",  
  "ratingParameter": {  
    "name": "unit",  
    "value": "minutes"  
  },  
  "volume": "10"  
},  
"referenceCode": "REF-12345",  
"referenceSequence": "2",  
"transactionOperationStatus": "Charged"  
}}

Response:
D.37 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.38 Get amount converted from volume (section 5.20.3.1)

Request:

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

PLAN-001"&currency="USD"
Accept: application/json
Host: example.com:80
Appendix E. Parlay X operations mapping (Informative)

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

<table>
<thead>
<tr>
<th>ParlayREST Resource</th>
<th>ParlayREST method</th>
<th>ParlayREST section reference</th>
<th>Parlay X equivalent operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>All amount charge and refund transactions for an end user</td>
<td>POST</td>
<td>5.5.5</td>
<td>chargeAmount and refundAmount</td>
</tr>
<tr>
<td>All amount split charge transactions for an end user</td>
<td>POST</td>
<td>5.6.5</td>
<td>chargeSplitAmount</td>
</tr>
<tr>
<td>All volume charge and refund transactions for an end user</td>
<td>POST</td>
<td>5.9.5</td>
<td>chargeVolume and refundVolume</td>
</tr>
<tr>
<td>All volume split charge transactions for an end user</td>
<td>POST</td>
<td>5.10.5</td>
<td>chargeSplitVolume</td>
</tr>
<tr>
<td>All amount reservation transactions for an end user</td>
<td>POST</td>
<td>5.15.5</td>
<td>reserveAmount</td>
</tr>
<tr>
<td>Individual amount reservation transaction for an end user</td>
<td>POST</td>
<td>5.16.5</td>
<td>reserveAdditionalAmount, chargeReservation and releaseReservation</td>
</tr>
<tr>
<td>All volume reservation transactions for an end user</td>
<td>POST</td>
<td>5.17.5</td>
<td>reserveVolume</td>
</tr>
<tr>
<td>Individual volume reservation transaction for an end user</td>
<td>POST</td>
<td>5.18.5</td>
<td>reserveAdditionalVolume, chargeReservation and releaseReservation</td>
</tr>
<tr>
<td>Amount converted from given volume</td>
<td>GET</td>
<td>5.20.3</td>
<td>getAmount</td>
</tr>
</tbody>
</table>
Appendix F. Mapping of ParlayREST Reservation Transactions to the Charging Enabler (Informative)

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

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

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

F.1 Charging Mechanism

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

The three phases of a Charging session are as follows:

1. In a first step (first interrogation), a Reserve Units operation is performed. This maps 1:1 to the reserving operation “nil → reserved”. The application acting as a Charging Enabler User requests a number of units (money, data volume, usage time etc.), called RSU (Requested Service Units), from the Charging Enabler. At the end of this step, the Charging Enabler has deducted a certain number of units from the user’s account, and has granted them to the Charging Enabler User for consumption. These units are named GSU (Granted Service Units).

2. In subsequent steps (intermediate interrogation), the ReserveAndDebit Units operation is used by the application to debit consumed units and to make new reservations in one operation. Such ReserveAndDebit Units request includes the number of Used Service Units (USU) for debiting and a number of Requested Service Units (RSU) for subsequent consumption. Note that all GSU from a previous step are rendered invalid at this point in time. The Charging Enabler debits the USU to the user’s account, and returns a new GSU to the application for consumption. This step can be performed multiple times.

3. In a last step (final interrogation), the last debiting of Used Service Units (USU) is invoked by the application with the Debit Units operation. The Charging Enabler debits the USUs, puts the remaining GSUs back into the user’s account and closes the session.

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

F.2 Detailed Mapping of ParlayREST to Charging Enabler

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

This section describes a possible realization of the transaction operation state transitions in ParlayREST (see section 5.2.13) based on those Charging Request and Charging Response message pairs.
In order to be able to use a unified pattern for amount and volume charging, the following conventions are used below: The variable $amountVolumeReserved$ corresponds to the ParlayREST elements $amountReserved$ and $volumeReserved$. The variable $totalAmountVolumeCharged$ corresponds to the ParlayREST elements $totalAmountCharged$ and $totalVolumeCharged$. The variable $amountVolume$ corresponds to the ParlayREST elements $paymentAmount/amount$ and $paymentVolume/volume$. The parameters GSU, RSU and USU are as defined above.

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

1) initial reserving as part of ParlayREST resource creation

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

\[
RSU = amountVolume \\
\text{send Charging Request(INITIAL, RSU) and receive Charging Response(INITIAL, GSU) } \\
\text{amountVolumeReserved} = GSU
\]

2) charging as ParlayREST resource update

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

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

3) reserving as ParlayREST resource update

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

\[
RSU = amountVolumeReserved + amountVolume \\
USU = 0 \\
\text{send Charging Request(UPDATE, USU, RSU) and receive Charging Response(UPDATE, GSU) } \\
\text{amountVolumeReserved} = GSU
\]

4) releasing as ParlayREST resource update

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

\[
USU = 0 \\
\text{send Charging Request(TERMINATION, USU) and receive Charging Response(TERMINATION, USU) } \\
\text{amountVolumeReserved} = 0
\]

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

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