RESTful Network API for Payment

Approved Version 1.0.1 – 23 Oct 2015

Open Mobile Alliance

OMA-TS-REST_NetAPI_Payment-V1_0_1-20151023-A

Contents

Figures

Tables

1. Scope

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

2. References

2.1 Normative References

[3GPP 29.199-6]	3GPP Technical Specification, "Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (Release 8)", URL:http://www.3gpp.org/
[Autho4API_10]	"Authorization Framework for Network APIs", Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL:http://www.openmobilealliance.org/
[ISO4217]	"ISO 4217 currency names and code elements", URL:http://www.iso.org/
[REST_NetAPI_ACR]	"RESTful Network API for Anonymous Customer Reference Management", Open Mobile Alliance™, OMA-TS-REST_NetAPI_ACR-V1_0, URL:http://www.openmobilealliance.org/
[REST_NetAPI_Common]	"Common definitions for RESTful Network APIs", Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL:http://www.openmobilealliance.org/
[REST_NetAPI_Notification Channel]	"RESTful Network API for Notification Channel", Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, URL:http://www.openmobilealliance.org/
[REST_SUP_Payment]	"XML schema for the RESTful Network API for Payment", Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_payment-V1_0, URL:http://www.openmobilealliance.org/
[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt
[RFC2616]	"Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding et. al, January 1999, URL:http://www.ietf.org/rfc/rfc2616.txt
[RFC3261]	"SIP: Session Initiation Protocol", J. Rosenberg et al., June 2002, URL:http://www.rfc-editor.org/rfc/rfc3261.txt
[RFC3966]	"The tel URI for Telephone Numbers", H.Schulzrinne, December 2004, URL:http://www.ietf.org/rfc/rfc3966.txt
[RFC3986]	"Uniform Resource Identifier (URI): Generic Syntax", R. Fielding et. al, January 2005, URL:http://www.ietf.org/rfc/rfc3986.txt
[RFC4234]	"Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt
[RFC4627]	"The application/json Media Type for JavaScript Object Notation (JSON)", D. Crockford, July 2006, URL:http://www.ietf.org/rfc/rfc4627.txt
[SCRRULES]	"SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/
[W3C_URLENC]	HTML 4.01 Specification, Section 17.13.4 Form content types, The World Wide Web Consortium, URL:http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1
[XMLSchema1]	W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL:http://www.w3.org/TR/xmlschema-1/
[XMLSchema2]	W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL:http://www.w3.org/TR/xmlschema-2/

2.2   Informative References

[CHRG_TS_ONLINE]	"OMA Online Charging Interface", Version 1.1, Open Mobile Alliance™. OMA-TS-Charging_Online-V1_1, URL:http://www.openmobilealliance.org/
[OMADICT]	"Dictionary for OMA Specifications", Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, URL:http://www.openmobilealliance.org/
[ParlayREST_Payment]	"RESTful bindings for Parlay X Web Services – Payment", Version 1.1, Open Mobile Alliance™, OMA-TS-ParlayREST_Payment-V1_1
[REST_WP]	"Guidelines for RESTful Network APIs", Open Mobile Alliance™, OMA-WP-Guidelines_for_RESTful_Network_APIs, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

Client-side Notification URL	An HTTP URL exposed by a client, on which it is capable of receiving notifications and that can be used by the client when subscribing to notifications.
Notification Channel	A channel created on the request of the client and used to deliver notifications from a server to a client. The channel is represented as a resource and provides means for the server to post notifications and for the client to receive them via specified delivery mechanisms.
Notification Server	A server that is capable of creating and maintaining Notification Channels.
Server-side Notification URL	An HTTP URL exposed by a Notification Server, that identifies a Notification Channel and that can be used by a client when subscribing to notifications.

3.3 Abbreviations

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

4. Introduction

The Technical Specification for the RESTful Network API for Payment contains the HTTP protocol binding based on Parlay X Payment Web Services [3GPP 29.199-6] specification, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON, and application/x-www-form-urlencoded).

4.1   Version 1.0

The RESTful Network API for Payment V1.0 is a republication of the ParlayREST Payment API V1.1 [ParlayREST_Payment] as part of the suite of OMA RESTful Network APIs.
Bug fixes and structural changes to fit that suite, but also functional changes have been applied.

Version 1.0 of the RESTful Network API for Payment keeps supporting the operations introduced in [ParlayREST_Payment], as there are:

- Charging an amount, split amount, volume or split volume to an end user's account

- Refunding an amount or volume to an end user' account

- Reserving an amount or volume for an end user's account

- Adding an amount or volume to an existing reservation

- Charging to a previously made reservation

- Releasing funds left in a previously made reservation

The following new functionality has been introduced:

- Support for scope values used with authorization framework defined in [Autho4API_10]

- Support for Anonymous Customer Reference (ACR) as an end user identifier

- Support for "acr:auth" as a reserved keyword in a resource URL variable that identifies an end user

- Support for asynchronous transactions (new transactionOperationStatus value "Processing")

- Implicit subscription to notifications about the status change of a transaction

- Notifications about the status change of a transaction

5. Payment API definition

This section is organized to support a comprehensive understanding of the RESTful 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_NetAPI_Common].

The remainder of this document is structured as follows:

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

Section 6 contains the detailed specification for each of the resources. Each such subsection defines the resource, the request URI variables that are common for all HTTP commands, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. Application/x-www-form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D.

Appendix B provides the Static Conformance Requirements (SCR).

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

Appendix F provides a list of all light-weight resources, where applicable.

Appendix G defines authorization aspects to control access to the resources defined in this specification.

Finally, Appendix H describes the mapping of RESTful Payment API Reservation Transactions to the OMA Online Charging Interface [CHRG_TS_ONLINE].

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

5.1   Resources Summary

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

The "apiVersion" URL variable SHALL have the value "v1" to indicate that the API corresponds to this version of the specification. See [REST_NetAPI_Common] which specifies the semantics of this variable.

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

```
//{serverRoot}
    └── / payment/{apiVersion}
            └── {endUserId}
                    ├── /transactions ▢
                    │       ├── /amount    ▢
                    │       │       └── /{transactionId} ▢
                    │       ├── /amountSplit  ▢
                    │       │       └── /{transactionId} ▢
                    │       ├── /volume    ▢
                    │       │       └── /{transactionId} ▢
                    │       │               └── /paymentAmount▢
                    │       ├── /volumeSplit  ▢
                    │       │       └── /{transactionId} ▢
                    │       │               └── /paymentAmount▢
                    │       ├── /amountReservation▢
                    │       │       └── /{transactionId} ▢
                    │       └── /volumeReservation▢
                    │               └── /{transactionId} ▢
                    └── / convertedVolume            └── /paymentAmount▢
                            └── /paymentAmount▢
```
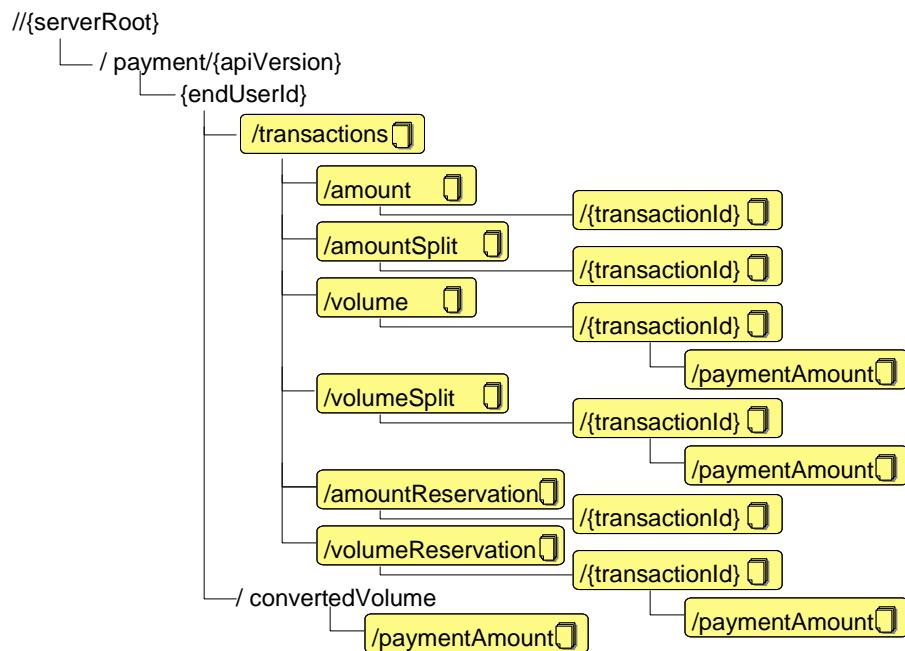
Figure 1 Resource structure defined by this specification

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

Purpose: to allow client to retrieve all payment transactions

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

Purpose: to allow client to perform amount charge, split charge and refund transactions

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

Resource	URL Base URL: http://{serverRoot}/payment/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
amount charge or refund transaction for an end user	unt/{transactionId}		transaction details (charge, refund)			
Individual amount split charge transaction for an end user	/{endUserId}/transactions/amountSplit/{transactionId}	AmountSplitTransaction	return amount split transaction details (charge)	no	no	no

Purpose: to allow client to perform volume charge, split charge and refund transactions

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

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

Purpose: to allow client to perform amount reservation transactions

Resource	URL **Base URL: http://{serverRoot}/payment/ {apiVersion}**	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
All amount reservation transactions for an end user	/{endUserId}/transactions/amountReservation	PaymentTransactionList (used for GET) AmountReservationTransaction (used for POST)	return all amount reservation transactions details for a	no	create new reservation transaction for a given end user	no

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

Purpose: to allow client to perform volume reservation transactions

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

Resource	URL **Base URL:** **http://{serverRoot}/payment/** **{apiVersion}**	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
			based on volume)			

Purpose: to allow client to obtain amount converted from volume transaction

Resource	URL **Base URL:** **http://{serverRoot}/payment/** **{apiVersion}**	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
Amount converted from given volume	/{endUserId}/convertedVolume /paymentAmount	PaymentAmount	return the amount resulting from converting the given volume	no	no	no

Purpose: to allow client to receive notifications about payment transactions

Resource	URL **<specified by the client>**	Data Structures	HTTP verbs			
			GET	**PUT**	**POST**	**DELETE**
Client notification about payment transaction	<specified by the client when transaction is created or during provisioning process>	PaymentTransactionNotification	no	no	notifies client about transaction status	no

5.2 Data Types

5.2.1 XML Namespaces

The namespace for the Payment data types is:

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

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

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

Note: Server implementations can choose to also support the legacy namespace urn:oma:xml:rest:payment:1 for the Payment data types, in order to allow backwards-compatibility with [ParlayREST_Payment] applications. Use of this legacy namespace is deprecated and support is foreseen to be withdrawn in future versions of this specification. In messages sent from the server to the application, the legacy namespace is suggested to be used by the server if it was used by a legacy application in the corresponding request or subscription message.

5.2.2 Structures

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

Some of the structures can be instantiated as so-called root elements.

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

5.2.2.1 Type: PaymentTransactionList

List of all payment transactions

Element	Type	Optional	Description
amountTransaction	AmountTransaction [0..unbounded]	Yes	Collection of AmountTransaction
amountSplitTransaction	AmountSplitTransaction [0..unbounded]	Yes	Collection of AmountSplitTransaction
volumeTransaction	VolumeTransaction [0..unbounded]	Yes	Collection of VolumeTransaction
volumeSplitTransaction	VolumeSplitTransaction [0..unbounded]	Yes	Collection of VolumeSplitTransaction
volumeReservationTransaction	VolumeReservationTransaction [0..unbounded]	Yes	Collection of VolumeReservationTransaction

amountReservationTran saction	AmountReser vationTransacti on [0..unbounded]	Yes	Collection of AmountReservationTransaction
resourceURL	xsd:anyURI	No	Self referring URL
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource.

A root element named paymentTransactionList of type PaymentTransactionList is allowed in response bodies.

5.2.2.2 Type: PaymentTransactionNotification

Payment transaction notification

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating a payment transaction. See [REST_NetAPI_Common] for details.
amountTransaction	AmountTransa ction	Choice	AmountTransaction
amountSplitTransaction	AmountSplitTr ansaction	Choice	AmountSplitTransaction
volumeTransaction	VolumeTransa ction	Choice	VolumeTransaction
volumeSplitTransaction	VolumeSplitTr ansaction	Choice	VolumeSplitTransaction
volumeReservationTran saction	VolumeReserv ationTransacti on	Choice	VolumeReservationTransaction
amountReservationTran saction	AmountReserv ationTransacti on	Choice	AmountReservationTransaction
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource.

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

5.2.2.3   Type: AmountTransaction

Transaction for charge or refund amount

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The identifier of the end user's account to be charged (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
paymentAmount	PaymentAmount	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc.
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request)
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client 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 duplicate transaction creation 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.
callbackReference	common:CallbackReference	Yes	Client's notification endpoint and parameters

| resourceURL | xsd:anyURI | Yes | Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests. |
| link | common:Link[0..unbounded] | Yes | Provided by the server and points to other resources that are in relationship with the current resource |

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

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

5.2.2.4   Type: AmountSplitTransaction

Transaction for charge or refund split amount

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

clientCorrelator	xsd:string	Yes	A correlator that the client 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 duplicate transaction creation 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.
callbackReference	common:CallbackReference	Yes	Client's notification endpoint and parameters
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource

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

Regarding the clientCorrelator field, the note in section 5.2.2.3 applies.

5.2.2.5   Type: VolumeTransaction

Transaction for charge or refund volume

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The identifier of the end user's account to be charged (e.g. 'sip' URI, 'tel' URI, 'acr' URI).
			If endUserId is also part of the request URL, the two MUST have the same value.
paymentVolume	PaymentVolume	No	Information on the volume charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. charged, refunded, etc.
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request)

originalServerReference Code	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate transaction creation 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.
callbackReference	common:Callb ackReference	Yes	Client's notification endpoint and parameters
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource

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

Regarding the clientCorrelator field, the note in section 5.2.2.3 applies.

5.2.2.6   Type: VolumeSplitTransaction

Transaction for charge or refund split volume

Element	Type	Optional	Description
endUserShare	EndUserShare [1..unbounded]	No	This end user's share of the split charge
paymentVolume	PaymentVolu me	No	Information on the volume charge to be made
transactionOperationSta tus	TransactionOp erationStatus	No	E.g. charged, refunded, etc.
referenceCode	xsd:string	No	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic

serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes (e.g. to be used in case of a subsequent refund request related to this charge request)
originalServerReference Code	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate transaction creation 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.
callbackReference	common:Callb ackReference	Yes	Client's notification endpoint and parameters
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource

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

Regarding the clientCorrelator field, the note in section 5.2.2.3 applies.

5.2.2.7   Type: AmountReservationTransaction

Transaction for reserve or release amount

Element	Type	Optional	Description
endUserId	xsd:anyURI	Yes	The identifier of the end user's account to be charged (e.g. 'sip' URI, 'tel' URI, 'acr' URI). It MUST be present in the initial reservation request. If endUserId is also part of the request URL, the two MUST have the same value.

paymentAmount	PaymentAmount	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. reserved, charged, released, etc.
referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change. 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). Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes
originalServerReferenceCode	xsd:string	Yes	This may be used to reconcile a current transaction request with an original transaction
clientCorrelator	xsd:string	Yes	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 duplicate transaction creation 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.
callbackReference	common:CallbackReference	Yes	Client's notification endpoint and parameters
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource

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

Regarding the clientCorrelator field, the note in section 5.2.2.3 applies.

5.2.2.8   Type: VolumeReservationTransaction

Transaction for reserve or release volume

Element	Type	Optional	Description
endUserId	xsd:anyURI	Yes	The identifier of the end user's account to be charged (e.g. 'sip' URI, 'tel' URI, 'acr' URI). It MUST be present in the initial reservation request. If endUserId is also part of the request URL, the two MUST have the same value.
paymentVolume	PaymentVolume	No	Information on the amount charge to be made
transactionOperationStatus	TransactionOperationStatus	No	E.g. reserved, charged, released, etc.
referenceSequence	xsd:integer	No	Sequential number generated by client application for every transaction state change. 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). Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, e.g. in case of disputes. Used for business logic, not for operational logic
serverReferenceCode	xsd:string	Yes	A unique reference to the request, provided by the server, and meaningful to the server's backend system for correlation purposes
originalServerReferenceCode	xsd:string	Yes	This may be used to reconcile a current transaction request with an original transaction.

clientCorrelator	xsd:string	Yes	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 duplicate transaction creation 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.
callbackReference	common:CallbackReference	Yes	Client's notification endpoint and parameters
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server and points to other resources that are in relationship with the current resource

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

Regarding the clientCorrelator field, the note in section 5.2.2.3 applies.

5.2.2.9   Type: PaymentAmount

Payment amount specific information

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

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

5.2.2.10   Type: PaymentVolume

Volume amount specific information

Element	Type	Optional	Description
billingText	xsd:string	No	Textual information to appear on the bill
volume	xsd:decimal	No	The volume to be charged
ratingParameter	RatingParameter [0..unbounded]	Yes	Parameters to use when performing rating ("unit", "contract", "service", "operation", etc). There is a maximum of one instance of each parameter type. Example, for the request "reserve 5 minutes of the gold video service", the volume part value is 5 and the parameters part has the following properties: • "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.

5.2.2.11   Type: ChargingMetaData

Information about charging data

Element	Type	Optional	Description
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by. This provides visibility to the true merchant in an aggregation scenario and is key in dealing with customer queries where an aggregator submits requests on behalf of another entity.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased. A standard list of category codes would be helpful together with the ability to extend as required. This provides multiple uses including correct taxation, service blocking and service based spending limits.

channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required. This is useful if the operator needs to interact with the subscriber to authorise the charge (advice of charge) and provides details on how such interaction should be done. Some payment show the value in centrally controlled advice of charge and the channel is an important data element.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system. It is important to know if the amount submitted has been pre taxed so the subscribers do not get double taxed on their bill.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies. Allows operators to track charges and group them based on the subscription service which they belong to. Also allows operators to block subscription payments if they have been requested to by the subscriber. For one off purchases, it allows the separate gathering of user consent and provides non repudiation for purchases.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased. This field could contain (for example) the short code of the service or an internal partner service ID and will allow the purchase to be tied directly back to their service catalog. This will be especially important in the US carriers to track payments to specific mobile campaigns.
productId	xsd:string	Yes	Combines with the service ID to uniquely 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.

5.2.2.12 Type: EndUserShare

Information for splitting charges between accounts

Element	Type	Optional	Description
endUserId	xsd:anyURI	No	The identifier of the end user's account to be charged (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
percent	xsd: integer	No	The percentage of this end user's share. The sum of all shares must equal 100. The value of percentage should be positive.

5.2.2.13 Type: RatingParameter

Rating information for volume payment

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

5.2.3 Enumerations

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

5.2.3.1 Enumeration: TransactionOperationStatus

Enumeration of supported status resulting from transaction operations

Enumeration	Description
Charged	In the request: charge the amount or volume.
	In the response:the amount or volume has been charged.
Refunded	In the request: refund the amount or volume.
	In the response:the amount or volume has been refunded.
Reserved	In the request: reserve the amount or volume.
	In the response:the amount or volume has been reserved.
Released	In the request: release the reservation.
	In the response:the reservation is released (implicitly by expiration or explicitly by release operation).
Denied	This value only occurs in responses to a GET operation and means that the operation has been denied by the back-end payment server because of an issue with the user's account (e.g. insufficient balance, potential fraud flag on the user's account).
Refused	This value only occurs in responses to a GET operation and means that the operation has been refused (i.e. rejected by end user).
Processing	This value only occurs in responses to POST or GET operations and means that the operation is not done yet (e.g. the charge has not been yet added, the reservation is still processing).

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 as answer to the POST request an HTTP status code from the 4xx range and a RequestError data structure, with an indication of the or PolicyException as defined in or referenced from section 7. 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.

An operation can be accepted for processing but the processing has not been completed, i.e. the transaction is not in its final status. In such a case the Server MAY return an HTTP 202 Accepted status, along with a representation of the resource, as answer to the POST request. In this response the transactionOperationStatus value SHALL be "Processing". The application can later check the status of the transaction using a GET operation on the resource or receive a notification of the final status of the transaction. The support of the "Processing" status is OPTIONAL for the server although the application MUST support this status value. If the server supports the "Processing" status, the server SHALL also support GET operations on the individual transaction resources and SHOULD support notifications of final status of individual transaction resources.

Note: The "Processing" status could be used when serving the request needs to run an external process whose execution time is expected to be too long (e.g. a charge or reserve operation requires out-of-band user's authorization). However, it is an implementation decision whether to use this status. A server might always use the "Processing" status for the charge and/or reserve operations, i.e. the behaviour is asynchronous by default. Alternatively, it might start an internal fixed timer when the charge or reserve operation arrives and only signal the "Processing" status when the result of the external process is not available within the timer value.

5.2.4 Values of the Link "rel" attribute

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

- PaymentTransactionList

- AmountTransaction

- AmountSplitTransaction

- AmountReservationTransaction

- VolumeTransaction

- VolumeSplitTransaction

- VolumeReservationTransaction

- PaymentAmount

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

5.3 Sequence diagrams

The following subsections describe the resources, methods and steps involved in typical scenarios.

In a sequence diagram, a step which involves delivering a notification is labeled with "POST or NOTIFY", where "POST" refers to delivery via the HTTP POST method, and "NOTIFY" refers to delivery using the Notification Channel [REST_NetAPI_NotificationChannel].

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}/payment/{apiVersion}/{endUserId}/transactions/amount
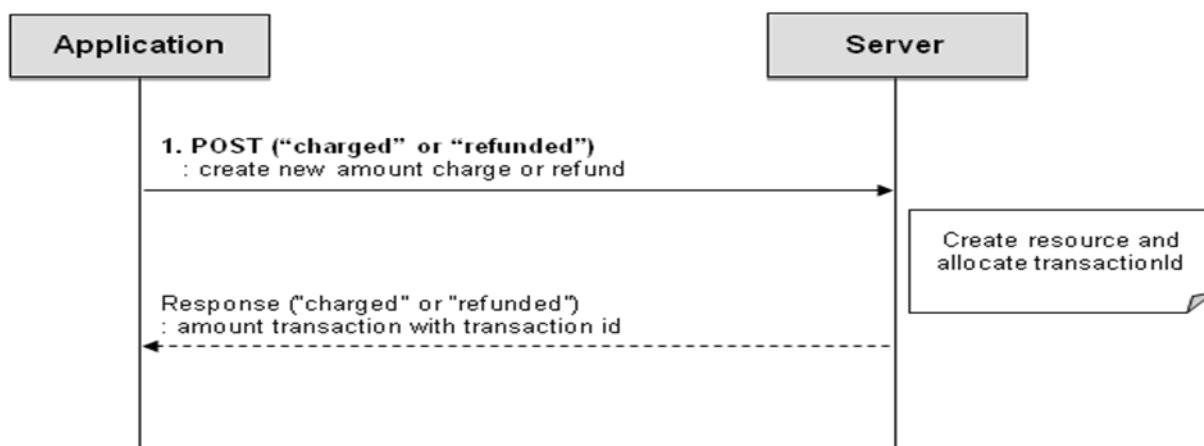
Figure 2 Amount charge and refund transaction

Outline of flow:

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

5.3.2 Amount split charge transaction

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

The resource:

− To create an amount split charge transaction for an end user, create new resource under
http://{serverRoot}/payment/{apiVersion}/{endUserId}/transactions/amountSplit
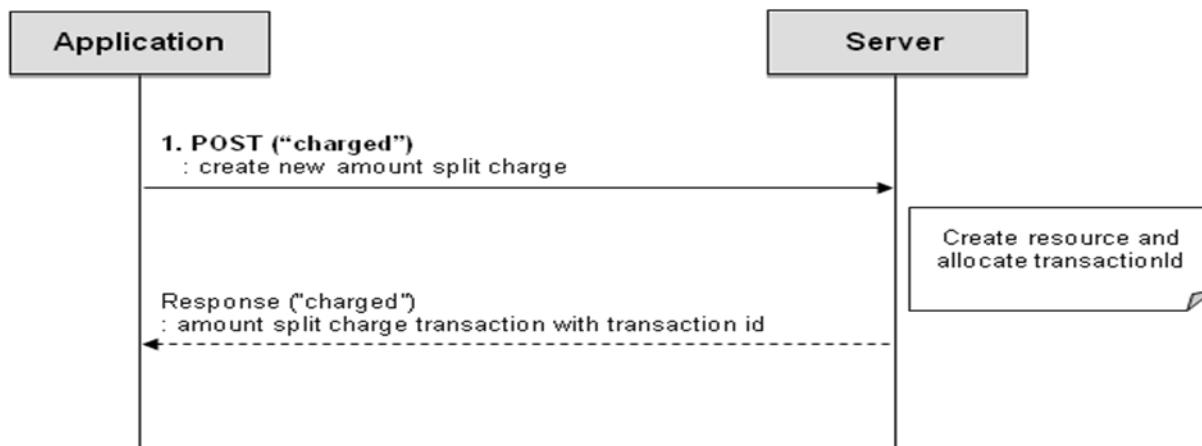
Figure 3 Amount split charge transaction

Outline of flow:

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

5.3.3   Volume charge and refund transaction

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

The resource:

− To create a volume charge or refund transaction for an end user, create new resource under
http://{serverRoot}/payment/{apiVersion}/{endUserId}/transactions/volume
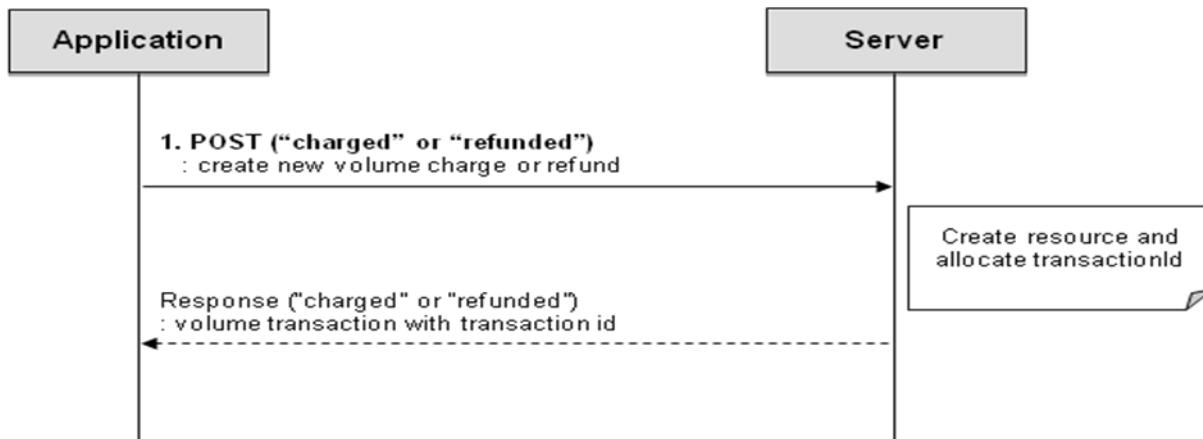
Figure 4 Volume charge and refund transaction

Outline of flow:

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

5.3.4   Volume split charge transaction

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

The resource and operation used

– To create a volume split charge transaction for an end user, create new resource under
http://{serverRoot}/payment/{apiVersion}/{endUserId}/transactions/volumeSplit
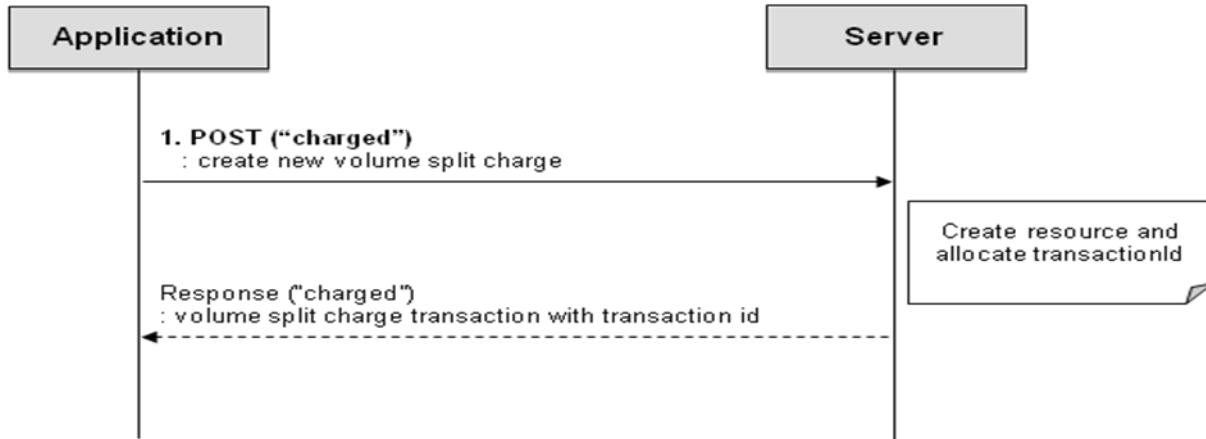
Figure 5 Volume split charge transaction

Outline of flow:

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

5.3.5   Amount reservation transaction

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

The resources:

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

– To update an amount reservation transaction for an end user, use resource
http://{serverRoot}/payment/{apiVersion}/{endUserId}/transactions/amountReservation/{transactionId}
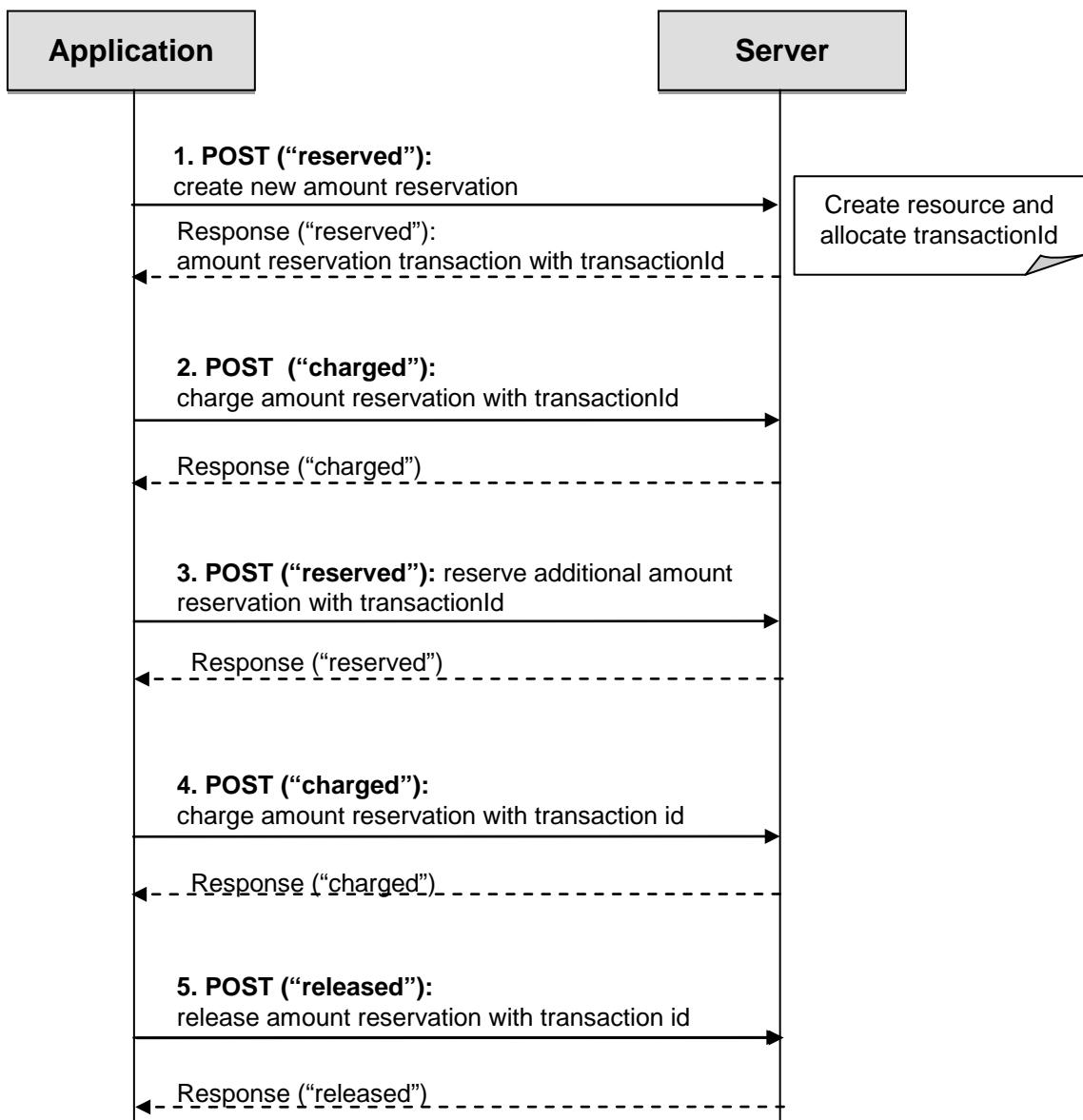
Figure 6 Amount reservation transaction

Outline of flow:

1. Application creates a new amount reservation transaction for an end user. (POST)

2. Application charges an amount reservation with transactionId. (POST)

3. In case that add to the existing amount reservation, application reserves an additional amount reservation with transactionId. (POST)

4. Application charges an amount reservation with transactionId. (POST)

5. Application releases an amount reservation transaction for an end user. (POST)

5.3.6   Volume reservation transaction

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

The resource:

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

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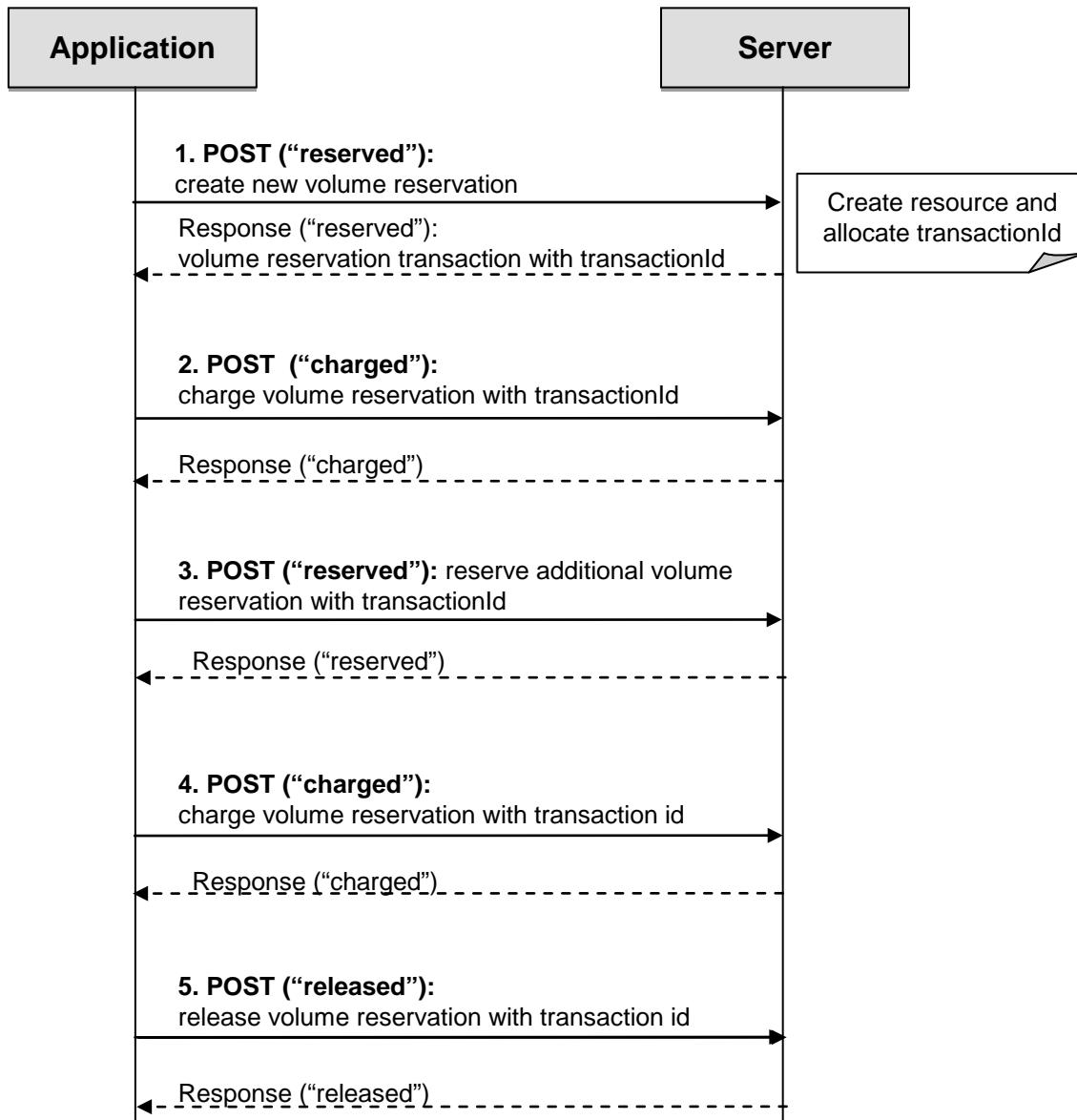
Figure 7 Volume reservation transaction

Outline of flow:

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

2. Application charges a volume reservation with transactionId. (POST)

3. In case that add to the existing volume reservation, application reserves an additional volume reservation with transactionId. (POST)

4. Application charges a volume reservation with transactionId. (POST)

5. Application releases a volume reservation transaction for an end user. (POST)

5.3.7 Amount converted from volume transaction

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

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

− To get the amount for a given volume use the resource below with mandatory URL parameter "volume" and applicable optional URL parameters (e.g. "unit", "contract", "service", "operation")
http://{serverRoot}/payment/{apiVersion}/{endUserId}/convertedVolume/paymentAmount
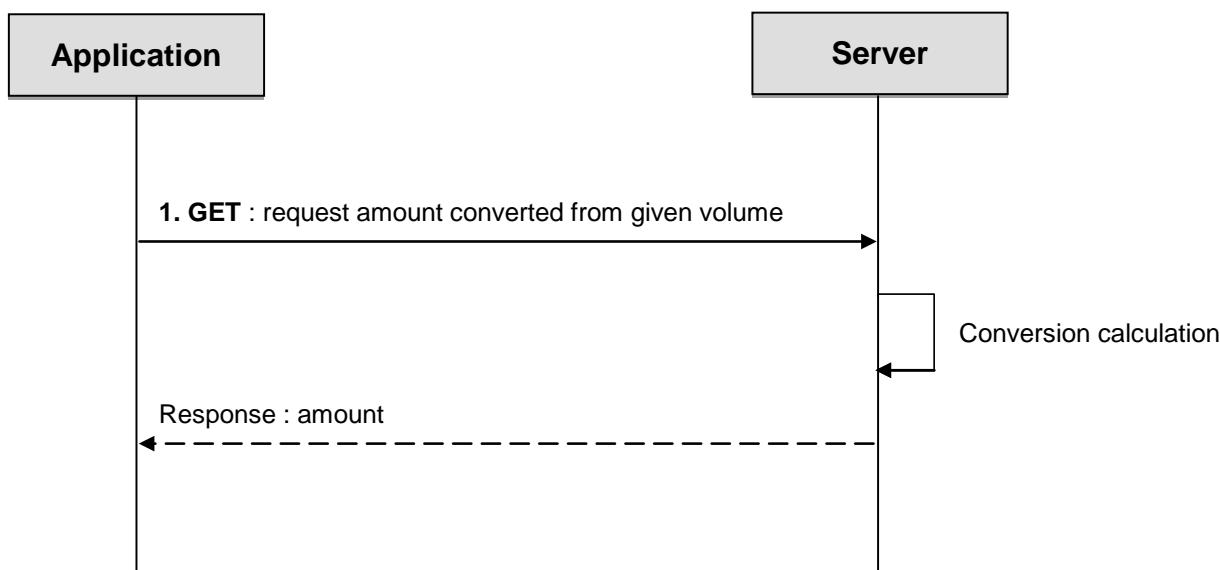
Figure 8 Amount converted from volume transaction

Outline of flow:

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

5.3.8 Charge or refund amount and check transaction status

This figure below shows a scenario for charging and amount and get the transaction status of the transaction.

The resources:

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

− To get the status of an amount charge or refund transaction for an end user, use resource
http://{serverRoot}/payment/{apiVersion}/{endUserId}/transactions/amount/{transactionId}
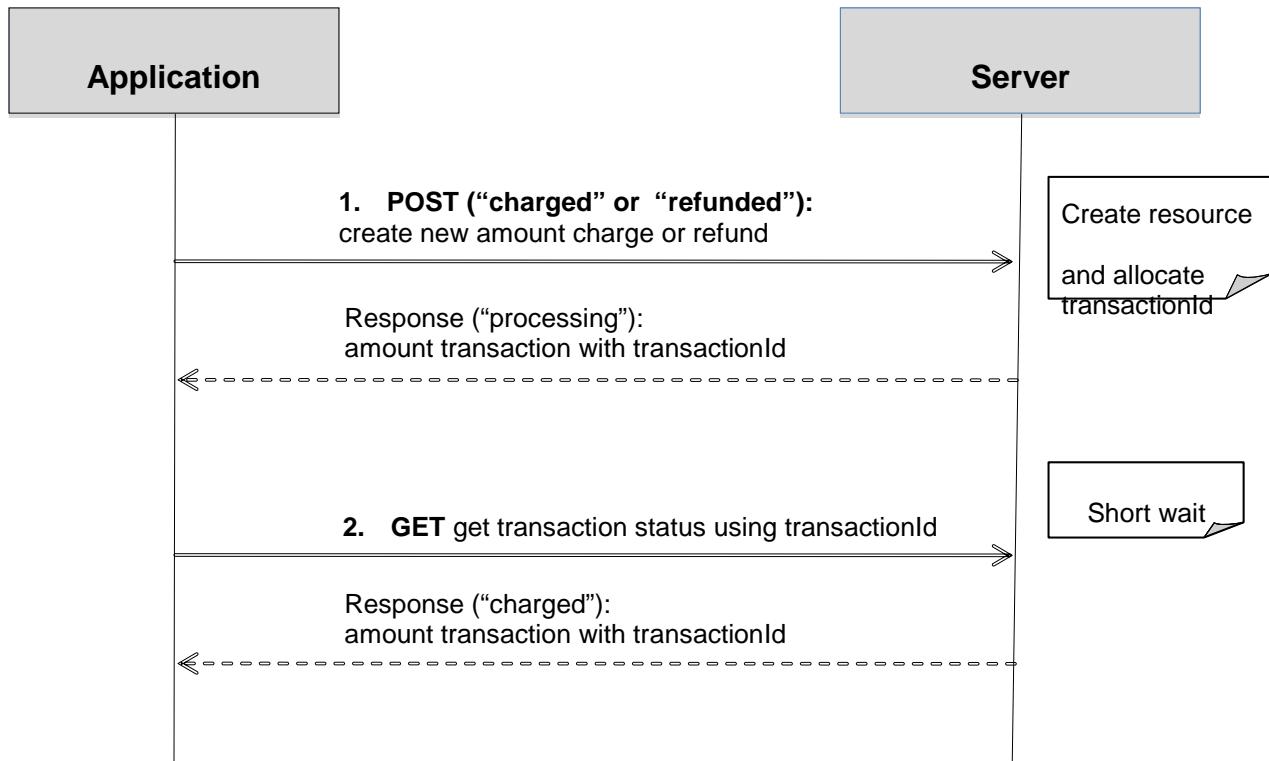
Figure 9 Amount transaction and status checking

Outline of the flows:

1. An application asks for the creation of new amount charge or refund transaction for an end user using POST and receives the response containing the amount transaction with a "Processing"status.

2. The application requests the transaction status with the given transactionId using GET and gets the final status.

5.3.9 Charge or refund amount and notify transaction status

This figure below shows a scenario for charging an amount and notifying the transaction status of the transaction.

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

The resources:

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

− To notify the application about the transaction status, POST notification to the client supplied notifyURL
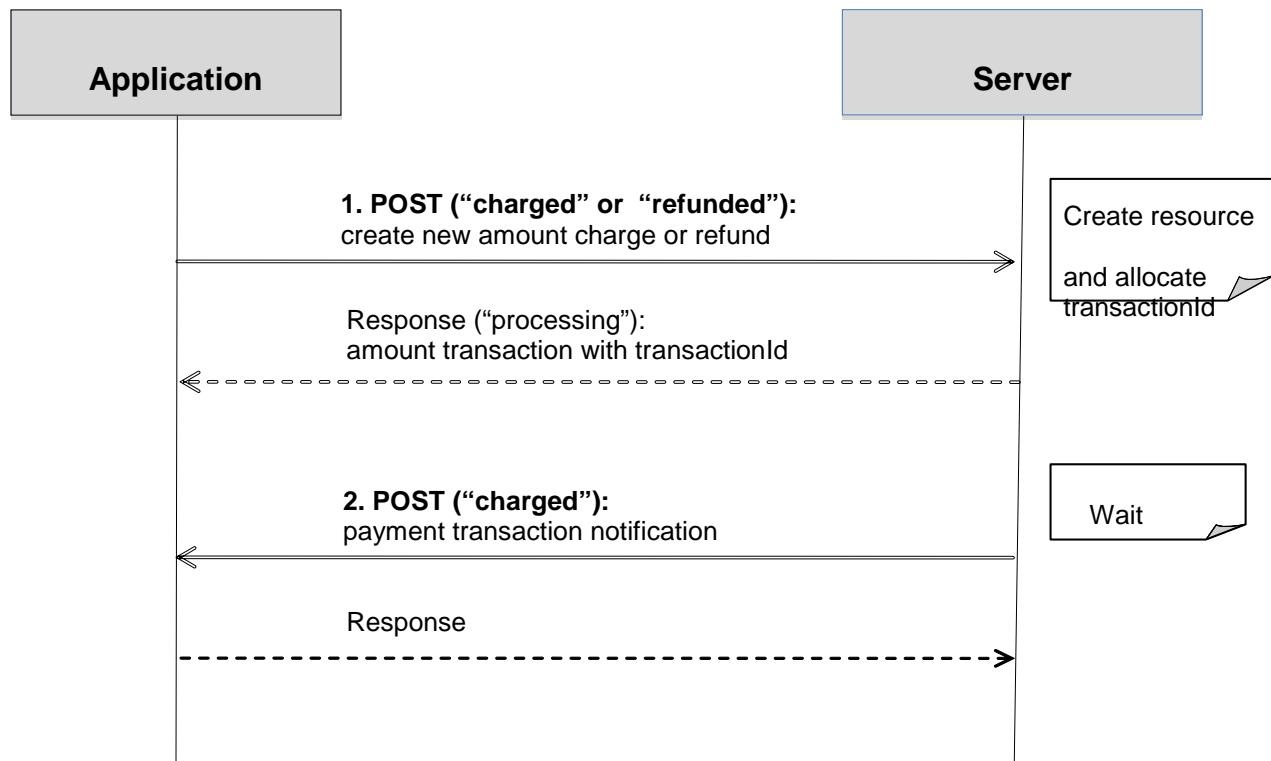
Figure 10 Amount transaction with notification

Outline of the flows:

1. An application asks for the creation of new amount charge or refund transaction for an end user using POST and provides a callbackReference for notifications. The application receives the response containing the amount transaction with a "Processing"status.

2. When the transaction reaches its final status, the server notifies the application the new transaction status.

6. Detailed specification of the resources

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

- Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) MUST be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in "resourceURL" and "link" elements).

- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an MSISDN, it MUST be defined as a global number according to [RFC3966] (e.g. tel:+19585550100). The use of characters other than digits and the leading "+" sign SHOULD be avoided in order to ensure uniqueness of the resource URL. This applies regardless of whether the user identifier appears in a URL variable or in a parameter in the body of an HTTP message.

- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of a SIP URI, it MUST be defined according to [RFC3261].

- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it MUST be defined according to Appendix H of [REST_NetAPI_ACR].

  o The ACR 'auth' is a supported reserved keyword, and MUST NOT be assigned as an ACR to any particular end user. See G.1.2 for details regarding the use of this reserved keyword.

- For requests and responses that have a body, the following applies: in the requests received, the server SHALL support JSON and XML encoding of the parameters in the body, and MAY support application/x-www-form-urlencoded parameters in the body. The Server SHALL return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST_NetAPI_Common]. In notifications to the Client, the server SHALL use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations SHALL follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST_ NetAPI_Common].

6.1 Resource: All payment transactions for an end user

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

6.1.1 Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.1.3   GET

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

6.1.3.1   Example 1: get all transactions   (Informative)

6.1.3.1.1   Request

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

6.1.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <!-- COMPLETED AMOUNT CHARGE TRANSACTION -->
 <amountTransaction>
  <endUserId>tel:+19585550100</endUserId>
    <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction  "Charged"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
   </chargingInformation>
   <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
 </amountTransaction>
 <!-- COMPLETED AMOUNT REFUND TRANSACTION -->
 <amountTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Refunded"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-456</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54322</clientCorrelator>
```

```
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans456</resourceURL>
</amountTransaction>
<!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
<amountSplitTransaction>
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>20</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</endUserId>
  <percent>80</percent>
 </endUserShare>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount transaction "Charged"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
 <serverReferenceCode>ABC-789</serverReferenceCode>
 <clientCorrelator>54323</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans789</resourceURL>
</amountSplitTransaction>
<!-- COMPLETED VOLUME CHARGE TRANSACTION -->
<volumeTransaction>
 <endUserId>tel:+19585550100</endUserId>
  <paymentVolume>
  <billingText>Test volume transaction "Charged"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>10</totalVolumeCharged>
 </paymentVolume>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
 <serverReferenceCode>DEF-123</serverReferenceCode>
 <clientCorrelator>55551</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124</resourceURL>
</volumeTransaction>
<!-- COMPLETED VOLUME REFUND TRANSACTION -->
<volumeTransaction>
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume transaction "Refunded"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeRefunded>10</totalVolumeRefunded>
```

```
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>GHI-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54324</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans125</resourceURL>
 </volumeTransaction>
 <!-- COMPLETED VOLUME SPLIT CHARGE TRANSACTION -->
 <volumeSplitTransaction>
  <endUserShare>
   <endUserId>tel:+19585550100</endUserId>
   <percent>20</percent>
  </endUserShare>
  <endUserShare>
   <endUserId>tel:+19585550101</endUserId>
   <percent>80</percent>
  </endUserShare>
    <paymentVolume>
   <billingText>Test volume transaction  "Charged"</billingText>
   <volume>10</volume>
   <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
   </ratingParameter>
   <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>JKL-123</serverReferenceCode>
  <clientCorrelator>54325</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans126</resourceURL>
 </volumeSplitTransaction>
 <!--
  VOLUME RESERVATION TRANSACTION "RESERVED"
 -->
 <volumeReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentVolume>
   <billingText>Test volume reservation transaction "Reserved"</billingText>
   <volume>10</volume>
   <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
   </ratingParameter>
   <totalVolumeCharged>0</totalVolumeCharged>
   <volumeReserved>10</volumeReserved>
  </paymentVolume>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>66666</clientCorrelator>

<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999</resourceURL>
 </volumeReservationTransaction>
 <!-- VOLUME RESERVATION TRANSACTION "CHARGED"-->
```

```
 <volumeReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
    <paymentVolume>
    <billingText>Test volume reservation transaction "Charged"</billingText>
    <volume>10</volume>
    <ratingParameter>
     <name>unit</name>
     <value>minutes</value>
    </ratingParameter>
    <totalVolumeCharged>25</totalVolumeCharged>
    <volumeReserved>0</volumeReserved>
   </paymentVolume>
   <transactionOperationStatus>Charged</transactionOperationStatus>
   <referenceSequence>4</referenceSequence>
   <referenceCode>REF-12345</referenceCode>
   <serverReferenceCode>MNO-123</serverReferenceCode>
   <clientCorrelator>66667</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans127</resourceURL>
 </volumeReservationTransaction>
 <!--
   COMPLETED AMOUNT RESERVATION TRANSACTION
 -->
 <amountReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount reservation transaction  "Charged"</description>
    <currency>USD</currency>
    <amount>15</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountCharged>25</totalAmountCharged>
   <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>PQR-123</serverReferenceCode>
  <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans128</resourceURL>
 </amountReservationTransaction>
 <!-- AMOUNT RESERVATION TRANSACTION "RESERVED"-->
 <amountReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <!-- last additional reserve applied -->
  <paymentAmount>
   <chargingInformation>
    <description>Test amount reservation transaction  "Reserved"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountCharged>0</totalAmountCharged>
   <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
```

```
  <referenceSequence>1</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>55556</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans998</resourceURL>
 </amountReservationTransaction>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions</resourceURL>
</payment:paymentTransactionList>
```

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

6.1.3.2.1 Request

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

6.1.3.2.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
 <link rel="PaymentTransactionList"
   href="http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions"/>
 <serviceException>
  <messageId>SVC0004</messageId>
  <text>No valid addresses provided in message part %1</text>
  <variables>endUserId=tel:+19585550100</variables>
 </serviceException>
</common:requestError>
```

6.1.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.1.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.1.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.2 Resource: All amount charge and refund transactions for an end user

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case,

the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.2.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.2.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.2.3   GET

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

6.2.3.1   Example: get all amount transactions   (Informative)

6.2.3.1.1   Request

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

6.2.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <!-- COMPLETED AMOUNT CHARGE TRANSACTION -->
 <amountTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Charged"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
   </chargingInformation>
   <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
```

```
   <transactionOperationStatus>Charged</transactionOperationStatus>
   <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
 </amountTransaction>
 <!-- COMPLETED AMOUNT CHARGE TRANSACTION - ACRed end user id-->
 <amountTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Charged"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>DEF-123</serverReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans124</resourceURL>
 </amountTransaction>
 <!-- COMPLETED AMOUNT REFUND TRANSACTION -->
 <amountTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Refunded"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>WXY-123</serverReferenceCode>
  <originalServerReferenceCode>DEF-123</originalServerReferenceCode>
  <clientCorrelator>54330</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans129</resourceURL>
 </amountTransaction>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount</resourceURL>
</payment:paymentTransactionList>
```

6.2.4   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 [RFC2616].

6.2.5   POST

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

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

[REST_NetAPI_NotificationChannel]). Example in section 6.2.5.2 illustrates the use of optional callbackReference in the request.

6.2.5.1   Example 1: create charge amount using 'tel' URI   (Informative)

6.2.5.1.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.2.5.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount transaction "Charged"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

6.2.5.2   Example 2: create charge amount with callbackReference and "Processing" status returned (Informative)

6.2.5.2.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount transaction "Processing"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>54321</clientCorrelator>
 <callbackReference>
    <notifyURL>http://application.example.com/notifications/PaymentTransactionFinalResult</notifyURL>
    <callbackData>12345</callbackData>
 </callbackReference>
</payment:amountTransaction>
```

6.2.5.2.2   Response

```
HTTP/1.1 202 Accepted
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount transaction "Processing"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
 </paymentAmount>
 <transactionOperationStatus>Processing</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
 <serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
```

```
<callbackReference>
  <notifyURL>http://application.example.com/notifications/PaymentTransactionFinalResult</notifyURL>
  <callbackData>12345</callbackData>
</callbackReference>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

6.2.5.3   Example 3: create refund amount   (Informative)

6.2.5.3.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
  </paymentAmount>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
```

6.2.5.3.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans130
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction "Refunded"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
```

```
<transactionOperationStatus>Refunded</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>XYZ-123</serverReferenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans130</resourceURL>
</payment:amountTransaction>
```

6.2.5.4   Example 4: 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_NetAPI_Common] for other possible cases.

6.2.5.4.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.2.5.4.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount transaction "Charged" </description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
```

```
 <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

6.2.5.5   Example 5: 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.

6.2.5.5.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.2.5.5.2   Response

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

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

6.2.5.6   Example 6: create charge amount using 'acr' URI   (Informative)

6.2.5.6.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>acr:pseudonym123</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>
```

6.2.5.6.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>acr:pseudonym123</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount transaction "Charged"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

6.2.6   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 [RFC2616].

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

The resource used is:

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.3.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.3.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.3.3   GET

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

6.3.3.1   Example: get all amount split transactions   (Informative)

6.3.3.1.1   Request

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

6.3.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
 <amountSplitTransaction>
  <endUserShare>
   <endUserId>tel:+19585550100</endUserId>
   <percent>30</percent>
  </endUserShare>
  <endUserShare>
   <endUserId>tel:+19585550101</endUserId>
   <percent>70</percent>
  </endUserShare>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Charged"</description>
    <currency>USD</currency>
    <amount>20</amount>
    <code>TEST-012345</code>
   </chargingInformation>
   <totalAmountCharged>20</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55552</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123</resourceURL>
 </amountSplitTransaction>
 <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
 <amountSplitTransaction>
  <endUserShare>
   <endUserId>tel:+19585550100</endUserId>
   <percent>20</percent>
  </endUserShare>
  <endUserShare>
   <endUserId>tel:+19585550101</endUserId>
   <percent>80</percent>
  </endUserShare>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount transaction "Charged" </description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012346</code>
   </chargingInformation>
   <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12346</referenceCode>
<serverReferenceCode>DEF-123</serverReferenceCode>
```

```
 <clientCorrelator>55553</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans124</resourceURL>
 </amountSplitTransaction>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit</resourceURL>
</payment:paymentTransactionList>
```

6.3.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.3.5   POST

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

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrates the use of optional callbackReference in the request

6.3.5.1   Example: create split charge amount   (Informative)

6.3.5.1.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT SPLIT CHARGE TRANSACTION -->
<payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>30</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</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>
```

6.3.5.1.2 Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT SPLIT CHARGE TRANSACTION -->
 <!-- COMPLETED AMOUNT SPLIT CHARGE TRANSACTION -->
<payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>30</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</endUserId>
  <percent>70</percent>
 </endUserShare>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount transaction "Charged" </description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54431</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123</resourceURL>
</payment:amountSplitTransaction>
```

6.3.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

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

The resource used is:

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

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

6.4.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.4.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.4.3   GET

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

6.4.3.1   Example: get amount charge (Informative)

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

6.4.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com
```

6.4.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount transaction "Charged" </description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
```

```
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

6.4.4 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 [RFC2616].

6.4.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.4.6 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 [RFC2616].

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

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

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

6.5.1 Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.5.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.5.3 GET

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

6.5.3.1   Example: get amount split charge   (Informative)

6.5.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123
HTTP/1.1
Accept: application/xml

Host: example.com
```

6.5.3.1.2   Response

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


<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT SPLIT CHARGE TRANSACTION -->
<payment:amountSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
   <endUserId>tel:+19585550100</endUserId>
   <percent>30</percent>
 </endUserShare>
 <endUserShare>
   <endUserId>tel:+19585550101</endUserId>
   <percent>70</percent>
 </endUserShare>
 <paymentAmount>
   <chargingInformation>
     <description>Test amount transaction "Charged" </description>
     <currency>USD</currency>
     <amount>10</amount>
     <code>TEST-012345</code>
   </chargingInformation>
   <totalAmountCharged>10</totalAmountCharged>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123</resourceURL>
</payment:amountSplitTransaction>
```

6.5.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.5.5   POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.5.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

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

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.6.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.6.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.6.3   GET

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

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

6.6.3.1.1   Request

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

6.6.3.1.2   Response

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

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

 <!-- COMPLETED VOLUME CHARGE TRANSACTION -->
 <volumeTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentVolume>
   <billingText>Test volume transaction "Charged" </billingText>
   <volume>10</volume>
   <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
   </ratingParameter>
   <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123</resourceURL>
 </volumeTransaction>
 <!-- COMPLETED VOLUME REFUND TRANSACTION -->
 <volumeTransaction>
  <endUserId>tel:+19585550100</endUserId>
    <paymentVolume>
   <billingText>Test volume transaction "Refunded"</billingText>
   <volume>10</volume>
   <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
   </ratingParameter>
   <totalVolumeRefunded>10</totalVolumeRefunded>
  </paymentVolume>
  <transactionOperationStatus>Refunded</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>DEF-123</serverReferenceCode>
  <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
  <clientCorrelator>55552</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124</resourceURL>
 </volumeTransaction>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume</resourceURL>
 </payment:paymentTransactionList>
```

6.6.4   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 [RFC2616].

6.6.5 POST

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

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

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

6.6.5.1.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.6.5.1.2 Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123
Content-Type: application/xml
Content-Length: nnnn


<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
   <billingText>Test volume transaction "Charged" </billingText>
   <volume>10</volume>
   <ratingParameter>
     <name>unit</name>
     <value>minutes</value>
   </ratingParameter>
```

```
   <totalVolumeCharged>10</totalVolumeCharged>
  </paymentVolume>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55551</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123</resourceURL>
 </payment:volumeTransaction>
```

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

6.6.5.2.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.6.5.2.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans12
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123</resourceURL>
</common:resourceReference>
```

6.6.5.3   Example 3: create refund volume   (Informative)

6.6.5.3.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME REFUND TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume transaction "Refunded"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
 </paymentVolume>
 <transactionOperationStatus>Refunded</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
 <clientCorrelator>55552</clientCorrelator>
</payment:volumeTransaction>
```

6.6.5.3.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME REFUND TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume transaction "Refunded"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeRefunded>10</totalVolumeRefunded>
 </paymentVolume>
 <transactionOperationStatus>Refunded</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
 <serverReferenceCode>DEF-123</serverReferenceCode>
 <originalServerReferenceCode>ABC-123</originalServerReferenceCode>
 <clientCorrelator>55552</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124</resourceURL>
</payment:volumeTransaction>
```

6.6.6   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 [RFC2616].

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

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.7.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123

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

6.7.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.7.3   GET

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

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

6.7.3.1.1   Request

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

6.7.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <volumeSplitTransaction>
  <endUserShare>
   <endUserId>tel:+19585550100</endUserId>
   <percent>20</percent>
  </endUserShare>
  <endUserShare>
   <endUserId>tel:+19585550101</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://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123</resourceURL>
 </volumeSplitTransaction>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit</resourceURL>
 <!-- COMPLETED VOLUME SPLIT CHARGE TRANSACTION -->
</payment:paymentTransactionList>
```

6.7.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.7.5   POST

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

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

6.7.5.1   Example: create volume split charge   (Informative)

6.7.5.1.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME SPLIT CHARGE TRANSACTION -->
<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>20</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</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>
```

6.7.5.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME SPLIT CHARGE TRANSACTION -->
<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>20</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</endUserId>
  <percent>80</percent>
 </endUserShare>
 <paymentVolume>
  <billingText>Test volume transaction "Charged" </billingText>
  <volume>10</volume>
```

```
 <ratingParameter>
  <name>unit</name>
  <value>minutes</value>
 </ratingParameter>
 </paymentVolume>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>55553</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123</resourceURL
>
</payment:volumeSplitTransaction>
```

6.7.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

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

The resource used is:

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

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

6.8.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.8.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.8.3   GET

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

6.8.3.1   Example: get volume charge   (Informative)

6.8.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.8.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME CHARGE TRANSACTION -->
<payment:volumeTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume transaction "Charged" </billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>10</totalVolumeCharged>
 </paymentVolume>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123</resourceURL></pay
ment:volumeTransaction>
```

6.8.4   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 [RFC2616].

6.8.5   POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.8.6   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 [RFC2616].

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

The resource used is:

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

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

6.9.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.9.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.9.3   GET

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

6.9.3.1   Example: get volume split charge   (Informative)

6.9.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.9.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME SPLIT CHARGE TRANSACTION -->
<payment:volumeSplitTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserShare>
  <endUserId>tel:+19585550100</endUserId>
  <percent>20</percent>
 </endUserShare>
 <endUserShare>
  <endUserId>tel:+19585550101</endUserId>
  <percent>80</percent>
 </endUserShare>  <paymentVolume>
  <billingText>Test volume transaction "Charged" </billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
```

```
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>10</totalVolumeCharged>
 </paymentVolume>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>55556</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123</resourceURL>
</payment:volumeSplitTransaction>
```

6.9.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.9.5   POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.9.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

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

The resource used is:

http://{serverRoot}/payment/{apiVersion}/{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.

6.10.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.10.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.10.3 GET

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

6.10.3.1 Example: get amount for volume charge   (Informative)

6.10.3.1.1 Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans999/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com
```

6.10.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
<chargingInformation>
 <description>10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+19585550100</description>
 <currency>USD</currency>
 <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

6.10.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.10.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.10.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

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

The resource used is:

http://{serverRoot}/payment/{apiVersion}/{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.

6.11.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.11.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.11.3   GET

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

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

6.11.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans999/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com
```

6.11.3.1.2   Response

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

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

6.11.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.11.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.11.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.12 Resource: All amount reservation transactions for an end user

The resource used is:

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.12.1 Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123

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

6.12.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.12.3 GET

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

6.12.3.1 Example: get all amount reservation transactions   (Informative)

6.12.3.1.1 Request

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

6.12.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <!-- COMPLETED AMOUNT RESERVATION TRANSACTION -->
 <amountReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>

  <paymentAmount>
   <chargingInformation>
    <description>Test amount reservation transaction "Charged" </description>
    <currency>USD</currency>
    <amount>15</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountCharged>25</totalAmountCharged>
   <amountReserved>0</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Charged</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <referenceCode>REF-12345</referenceCode>
  <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans123</resourc
eURL>
 </amountReservationTransaction>
 <!-- AMOUNT RESERVATION TRANSACTION "RESERVED" -->
 <amountReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
    <description>Test amount reservation transaction "Reserved"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
   </chargingInformation>
   <totalAmountCharged>0</totalAmountCharged>
   <amountReserved>10</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>1</referenceSequence>
  <clientCorrelator>55556</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourc
eURL>
 </amountReservationTransaction>
 <!--
   AMOUNT RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL RESERVATION AMOUNT APPLIED
 -->
 <amountReservationTransaction>
  <endUserId>tel:+19585550100</endUserId>
  <paymentAmount>
   <chargingInformation>
```

```
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>15</amount>
      <code>TEST012345</code>
     </chargingInformation>
     <totalAmountCharged>10</totalAmountCharged>
     <amountReserved>15</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans998</resourceURL>
   </amountReservationTransaction>
   <!--
    AMOUNT RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL
    RESERVATION AMOUNT IS APPLIED SECOND TIME
   -->
   <amountReservationTransaction>
    <endUserId>tel:+19585550100</endUserId>
    <paymentAmount>
     <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST012345</code>
     </chargingInformation>
     <totalAmountCharged>10</totalAmountCharged>
     <amountReserved>25</amountReserved>
    </paymentAmount>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>3</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans997</resourceURL>
   </amountReservationTransaction>
   <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation</resourceURL>
</payment:paymentTransactionList>
```

6.12.4   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 [RFC2616].

6.12.5   POST

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

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

6.12.5.1   Example: create amount reservation   (Informative)

6.12.5.1.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Reserved"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
 </paymentAmount>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <clientCorrelator>55555</clientCorrelator>
</payment:amountReservationTransaction>
```

6.12.5.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Reserved"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>10</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.12.6   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 [RFC2616].

6.13   Resource: Individual amount reservation transaction for an end user

The resource used is:

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

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

In the case an optional notification URL is passed to the server when updating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before updating the transaction.

6.13.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.13.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.13.3   GET

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

6.13.3.1   Example: get amount reservation   (Informative)

6.13.3.1.1   Request

```
GET /exampleAPI/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.13.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Reserved"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>10</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.13.5   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.

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

6.13.5.1   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.

6.13.5.1.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

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

```
<endUserId>tel:+19585550100</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>
```

6.13.5.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Charged" </description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
  <amountReserved>0</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.2 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 reques, and will not change the resource, instead just returning the same response as in the previous attempt.

6.13.5.2.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.13.5.2.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Charged" </description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
  <amountReserved>0</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>55555</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.3   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.

6.13.5.3.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn


<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Released"</description>
   <code>TEST012345</code>
  </chargingInformation>
 </paymentAmount>
 <transactionOperationStatus>Released</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
</payment:amountReservationTransaction>
```

6.13.5.3.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Released"</description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>0</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Released</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <clientCorrelator>55556</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.4   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.

6.13.5.4.1 Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>


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

6.13.5.4.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount reservation transaction "Charged" </description>
    <currency>USD</currency>
    <amount>5</amount>
    <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>5</totalAmountCharged>
  <amountReserved>5</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>55557</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.5   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.

6.13.5.5.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Released"</description>
   <code>TEST012345</code>
  </chargingInformation>
 </paymentAmount>
 <transactionOperationStatus>Released</transactionOperationStatus>
 <referenceSequence>3</referenceSequence>
</payment:amountReservationTransaction>
```

6.13.5.5.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Released"</description>
   <currency>USD</currency>
   <amount>5</amount>
   <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>5</totalAmountCharged>
  <amountReserved>0</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Released</transactionOperationStatus>
 <referenceSequence>3</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>55558</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.6 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.

6.13.5.6.1 Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.13.5.6.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
    <description>Test amount reservation transaction "Reserved"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST012345</code>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>15</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <clientCorrelator>55559</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

6.13.5.7 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.

6.13.5.7.1 Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.13.5.7.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
 <link rel="AmountReservationTransaction"
   href="http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999"/>
 <serviceException>
  <messageId>SVC0270</messageId>
  <text>Charging operation failed, the charge was not applied.</text>
 </serviceException>
</common:requestError>
```

6.13.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.14   Resource: All volume reservation transactions for an end user

The resource used is:

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

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

In the case an optional notification URL is passed to the server when creating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating the transaction.

6.14.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123

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

6.14.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.14.3   GET

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

6.14.3.1   Example: get all volume reservation transactions   (Informative)

6.14.3.1.1   Request

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

6.14.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionList xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <!-- VOLUME RESERVATION TRANSACTION "RESERVED" -->
 <volumeReservationTransaction>
```

```
    <endUserId>tel:+19585550100</endUserId>

    <paymentVolume>
     <billingText>Test volume reservation transaction "Reserved"</billingText>
     <volume>10</volume>
     <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
     </ratingParameter>
     <totalVolumeCharged>0</totalVolumeCharged>
     <volumeReserved>10</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>1</referenceSequence>
    <clientCorrelator>66666</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999</resourc
eURL>
   </volumeReservationTransaction>
   <!--
     VOLUME RESERVATION TRANSACTION "RESERVED" AFTER ADDITIONAL
     VOLUME IS APPLIED
   -->
   <volumeReservationTransaction>
    <endUserId>tel:+19585550100</endUserId>
    <paymentVolume>
     <billingText>Test volume reservation transaction "Reserved"</billingText>
     <volume>10</volume>
     <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
     </ratingParameter>
     <totalVolumeCharged>15</totalVolumeCharged>
     <volumeReserved>10</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Reserved</transactionOperationStatus>
    <referenceSequence>3</referenceSequence>
    <referenceCode>REF-12345</referenceCode>
    <clientCorrelator>66667</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans998</resourc
eURL>

   </volumeReservationTransaction>
   <!-- VOLUME RESERVATION TRANSACTION "CHARGED" -->
   <volumeReservationTransaction>
    <endUserId>tel:+19585550100</endUserId>
    <paymentVolume>
     <billingText>Test volume reservation transaction "Charged" </billingText>
     <volume>10</volume>
     <ratingParameter>
      <name>unit</name>
      <value>minutes</value>
     </ratingParameter>
     <totalVolumeCharged>25</totalVolumeCharged>
     <volumeReserved>0</volumeReserved>
    </paymentVolume>
    <transactionOperationStatus>Charged</transactionOperationStatus>
```

```
  <referenceSequence>4</referenceSequence>
  <referenceCode>REF-22345</referenceCode>
   <serverReferenceCode>ABC-123</serverReferenceCode>
  <clientCorrelator>66668</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans123</resourceURL>
  </volumeReservationTransaction>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation</resourceURL>
</payment:paymentTransactionList>
```

6.14.4 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 [RFC2616].

6.14.5 POST

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

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

6.14.5.1 Example 1: create reserve volume (Informative)

6.14.5.1.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME RESERVATION TRANSACTION -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.14.5.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME RESERVATION TRANSACTION  -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume reservation transaction "Reserved"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>0</totalVolumeCharged>
  <volumeReserved>10</volumeReserved>
 </paymentVolume>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <clientCorrelator>66666</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999</resourceURL>
</payment:volumeReservationTransaction>
```

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

6.14.5.2.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME RESERVATION TRANSACTION -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume reservation transaction "Reserved"</billingText>
  <volume>10</volume>
  <ratingParameter>
   <name>unit</name>
   <value>minutes</value>
  </ratingParameter>
 </paymentVolume>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <clientCorrelator>66667</clientCorrelator>
</payment:volumeReservationTransaction>
```

6.14.5.2.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC0004</messageId>
    <text>No valid addresses provided in message part %1</text>
    <variables>endUserId=tel:+19585550100</variables>
  </serviceException>
</common:requestError>
```

6.14.6   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 [RFC2616].

6.15   Resource: Individual volume reservation transaction for an end user

The resource used is:

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

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

In the case an optional notification URL is passed to the server when updating a transaction, this resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before updating the transaction.

6.15.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.15.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.15.3 GET

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

6.15.3.1 Example: get volume reservation   (Informative)

6.15.3.1.1 Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.15.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME RESERVATION TRANSACTION  -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
  <billingText>Test volume reservation transaction "Reserved"</billingText>
  <volume>10</volume>
  <ratingParameter>
    <name>unit</name>
    <value>minutes</value>
  </ratingParameter>
  <totalVolumeCharged>0</totalVolumeCharged>
  <volumeReserved>10</volumeReserved>
 </paymentVolume>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>66666</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999</resourceURL>
</payment:volumeReservationTransaction>
```

6.15.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.15.5 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.

The notifyURL in the optional callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]). Examples in this section do not illustrate the use of optional callbackReference in the request.

6.15.5.1   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.

6.15.5.1.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- VOLUME RESERVATION TRANSACTION  -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</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>
```

6.15.5.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
 <!-- VOLUME RESERVATION TRANSACTION  -->
<payment:volumeReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentVolume>
   <billingText>Test volume reservation transaction "Charged" </billingText>
   <volume>10</volume>
   <ratingParameter>
     <name>unit</name>
     <value>minutes</value>
   </ratingParameter>
   <totalVolumeCharged>10</totalVolumeCharged>
   <volumeReserved>0</volumeReserved>
 </paymentVolume>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceSequence>2</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>66666</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999</resourceURL>
```

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

6.15.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

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

The resource used is:

http://{serverRoot}/payment/{apiVersion}/{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.

6.16.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123
transactionId	unique transaction identifier

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

6.16.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.16.3   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.

6.16.3.1   Example: get amount for volume reservation   (Informative)

6.16.3.1.1   Request

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999/paymentAmount HTTP/1.1
Accept: application/xml
Host: example.com
```

6.16.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
<chargingInformation>
 <description>10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+19585550100</description>
 <currency>USD</currency>
 <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

6.16.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.16.5   POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.16.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC2616].

6.17   Resource: Amount converted from given volume

The resource used is:

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

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

6.17.1   Request URL variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
endUserId	the end user's account identifier. Examples:tel:+19585550100, acr:pseudonym123

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

6.17.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Payment, see [3GPP 29.199-6] and section 7 of the present document.

6.17.3   GET

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

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

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

6.17.3.1   Example: get amount converted from volume   (Informative)

6.17.3.1.1   Request

```
GET
/exampleAPI/payment/v1/tel%3A%2B19585550100/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MONTHL
Y-PLAN-001" HTTP/1.1
Accept: application/xml
Host: example.com
```

6.17.3.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentAmount xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
<chargingInformation>
   <description>100 Minutes converted to USD for endUserId=tel:+19585550100 according to contract="MONTHLY-PLAN-
001"</description>
   <currency>USD</currency>
   <amount>10</amount>
</chargingInformation>
</payment:paymentAmount>
```

6.17.4   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 [RFC2616].

6.17.5   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 [RFC2616].

6.17.6   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 [RFC2616].

6.18   Resource: Client notification about payment transaction

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

Note: In the case when the client has set up a Notification Channel to obtain the notifications, the client needs to use the mechanisms described in [REST_NetAPI_NotificationChannel], instead of the mechanism described below in section 6.18.5.

6.18.1   Request URL variables

Client provided.

6.18.2   Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

6.18.3   GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC2616].

6.18.4   PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC2616].

6.18.5   POST

This operation is used to notify client that a payment transaction status changed from a "Processing" status to a final status.

6.18.5.1   Example: Payment notification   (Informative)

6.18.5.1.1   Request

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

<?xml version="1.0" encoding="UTF-8"?>
<payment:paymentTransactionNotification xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <amountTransaction>
  <endUserId>tel:+19585550100</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>
  <clientCorrelator>54321</clientCorrelator>
 </amountTransaction>
</payment:paymentTransactionNotification>
```

6.18.5.1.2   Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:52:00 GMT
```

6.18.6   DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC2616].

7. Fault definitions

7.1 Service Exceptions

For common Service Exceptions refer to [REST_NetAPI_Common]. The following additional Service Exception codes are defined for the Payment API.

7.1.1 SVC0270: Charge failed

Name	Description
MessageID	SVC0270
Text	Charging operation failed, the charge was not applied.
Variables	None
HTTP status code(s)	403 Forbidden

7.1.2 SVC0271: Invalid sum of percentage allocations

Name	Description
MessageID	SVC0271
Text	Sum of percentage allocations is not equal to 100
Variables	None
HTTP status code(s)	400 Bad request

7.2 Policy Exceptions

For common Policy Exceptions refer to [REST_NetAPI_Common]. The following additional Policy Exception codes are defined for the Payment API.

7.2.1 POL0250: Too many endUserIdentifiers

Name	Description
MessageID	POL0250
Text	Too many end user identifiers are specified in message part %1
Variables	%1 - message part
HTTP status code(s)	403 Forbidden

7.2.2 POL0251: Split charging not supported

Name	Description
MessageID	POL0251
Text	Split Charging is not supported
Variables	None
HTTP status code(s)	403 Forbidden

7.2.3   POL0252: Refund request failed

In [ParlayREST_Payment], the POL0252 exception has been introduced to represent a number of different errors when creating refunds.

In the current specification, POL0252 has been superseded by POL1003 to POL1007, and the POL0252 exception SHALL NOT be returned by an API server conforming to the current specification.

7.2.4   POL0253: Payment operation refused by user

Name	Description
MessageID	POL0253
Text	Payment operation not authorised by user.
Variables	None
HTTP status code(s)	403 Forbidden

7.2.5   POL0254: Chargeable amount exceeded

Name	Description
MessageID	POL0254
Text	The amount exceeds the operator limit for a single charge,
Variables	None
HTTP status code(s)	403 Forbidden

7.2.6   POL1000: Insufficient credit

Name	Description
MessageID	POL1000
Text	User has insufficient credit for transaction
Variables	None
HTTP status code(s)	403 Forbidden

7.2.7   POL1001: Periodic charge threshold exceeded

Name	Description
MessageID	POL1001
Text	The %1 operator charging limit for this user has been exceeded
Variables	%1 – the time period (daily, weekly, monthly etc.)
HTTP status code(s)	403 Forbidden

7.2.8   POL1002: Charge velocity exceeded

Name	Description
MessageID	POL1002
Text	The charge happened too soon after the previous one.
Variables	None
HTTP status code(s)	403 Forbidden

7.2.9   POL1003: Refund exceeds original charge amount

Name	Description
MessageID	POL1003
Text	The refund amount exceeds the original amount charged %1
Variables	%1 – the original amount charged
HTTP status code(s)	403 Forbidden

7.2.10   POL1004: Refund below minimum refundable amount

Name	Description
MessageID	POL1004
Text	The refund amount is less than then minimum refundable amount supported %1
Variables	%1 – the minimum refundable amount supported
HTTP status code(s)	403 Forbidden

7.2.11   POL1005: OriginalServerReferenceCode not provided

Name	Description
MessageID	POL1005
Text	A refund request requires the originalServerReferenceCode for the charge that is being refunded
Variables	None
HTTP status code(s)	400 Bad request

7.2.12   POL1006: OriginalServerReferenceCode not valid

Name	Description
MessageID	POL1006
Text	The originalServerReferenceCode is not valid
Variables	None
HTTP status code(s)	400 Bad request

7.2.13   POL1007: Refunds not supported

Name	Description
MessageID	POL1007
Text	Refunds not supported
Variables	None
HTTP status code(s)	403 Forbidden

7.2.14   POL1008: Payment declined

Name	Description
MessageID	POL1008
Text	Alternative payment (credit card / other non-operator payment) method was declined
Variables	None
HTTP status code(s)	403 Forbidden

Appendix A.   Change History   (Informative)

A.1   Approved Version History

Reference	Date	Description
OMA-TS-REST_NetAPI_Payment-V1_0-20130924-A	24 Sep 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0278-INP_REST_NetAPI_Payment_V1_0_ERP_for_Final_Approval
OMA-TS-REST_NetAPI_Payment-V1_0_1-20151023-A	23 Oct 2015	Status changed to Approved by TP TP Ref # OMA-TP-2015-0167-INP_REST_NetAPI_Payment_V1_0_1_ERP_for_Notification

Appendix B.   Static Conformance Requirements   (Normative)

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

B.1   SCR for REST.Payment Server

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

B.1.1   SCR for REST.Payment.Transactions Server

Item	Function	Reference	Requirement
REST-PAY-TRANS-S-001-O	Support for access to all completed and pending transactions	6.1	REST-PAY-TRANS-S-002-O
REST-PAY-TRANS-S-002-O	Obtain all completed and pending transactions - GET	6.1.3	

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

Item	Function	Reference	Requirement
REST-PAY-AMNT-TRANS-S-001-M	Support for access to all amount charge and refund transactions for an end-user	6.2	
REST-PAY-AMNT-TRANS-S-002-O	Obtain all amount charge and refund transactions for an end-user - GET	6.2.3	
REST-PAY-AMNT-TRANS-S-003-M	Create an amount charge or refund transaction for an end-user – POST (XML or JSON)	6.2.5	
REST-PAY-AMNT-TRANS-S-004-O	Create an amount charge transaction for an end-user – POST (application/x-www-form-urlencoded)	C.1	
REST-PAY-AMNT-TRANS-S-005-O	Create an amount refund transaction for an end-user – POST (application/x-www-form-urlencoded)	C.2	

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

Item	Function	Reference	Requirement
REST-PAY-SPLIT-TRANS-S-001-O	Support for access to all amount split charge and refund transactions for an end-user	6.3	REST-PAY-SPLIT-TRANS-S-003-O
REST-PAY-SPLIT-TRANS-S-002-O	Obtain all amount split charge transactions for an end-user - GET	6.3.3	
REST-PAY-SPLIT-TRANS-S-003-O	Create a new amount split charge transaction for an end-user - POST	6.3.5	

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

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-S-001-O	Support access to individual amount charge or refund for an end-user	6.4	REST-PAY-IND-AMNT-S-002-O
REST-PAY-IND-AMNT-S-002-O	Retrieve individual completed or pending amount transaction - GET	6.4.3	

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

Item	Function	Reference	Requirement
REST-PAY-IND-SPLIT-S-001-O	Support access to individual split amount charge for an end-user	6.5	REST-PAY-IND-SPLIT-S-002-O
REST-PAY-IND-SPLIT-S-002-O	Retrieve individual completed or pending split amount transaction - GET	6.5.3	

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

Item	Function	Reference	Requirement
REST-PAY-VOL-TRANS-S-001-O	Support for access to all volume charge and refund transactions for an end-user	6.6	REST-PAY-VOL-TRANS-S-003-O AND REST-PAY-VOL-TRANS-S-004-O
REST-PAY-VOL-TRANS-S-002-O	Obtain all volume charge and refund transactions for an end-user - GET	6.6.3	
REST-PAY-VOL-TRANS-S-003-O	Create a volume charge transaction for an end-user - POST	6.6.5	
REST-PAY-VOL-TRANS-S-004-O	Create a volume refund transaction for an end-user - POST	6.6.5	

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

Item	Function	Reference	Requirement
REST-PAY-SPLIT-VOL-TRANS-S-001-O	Support for access to all volume split charge and refund transactions for an end-user	6.7	REST-PAY-SPLIT-VOL-TRANS-S-003-O
REST-PAY-SPLIT-VOL-TRANS-S-002-O	Obtain all volume split charge transactions for an end-user - GET	6.7.3	
REST-PAY-SPLIT-VOL-TRANS-S-003-O	Create a new volume split charge transaction for an end-user - POST	6.7.5	

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

Item	Function	Reference	Requirement
REST-PAY-IND-VOL-S-001-O	Support access to individual volume charge or refund transaction for an end-user	6.8	REST-PAY-IND-VOL-S-002-O
REST-PAY-IND-VOL-S-002-O	Retrieve individual completed or pending transaction - GET	6.8.3	

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

Item	Function	Reference	Requirement
REST-PAY-IND-SPLIT-VOL-S-001-O	Support access to individual split volume charge for an end-user	6.9	REST-PAY-IND-SPLIT-VOL-S-002-O
REST-PAY-IND-SPLIT-VOL-S-002-O	Retrieve individual completed or pending split volume charge transaction - GET	6.9.3	

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

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-VOL-S-001-O	Support access to individual payment information based on volume transaction	6.10	REST-PAY-IND-AMNT-VOL-S-002-O
REST-PAY-IND-AMNT-VOL-S-002-O	Retrieve individual payment information based on volume transaction - GET	6.10.3	

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

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-SPLIT-VOL-S-001-O	Support access to individual payment information based on volume split transaction	6.11	REST-PAY-IND-AMNT-SPLIT-VOL-S-002-O
REST-PAY-IND-AMNT-SPLIT-VOL-S-002-O	Retrieve individual payment information based on volume split transaction - GET	6.11.3	

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

Item	Function	Reference	Requirement
REST-PAY-AMNT-RES-TRANS-S-001-M	Support for access to all completed and pending amount reservation transactions	6.12	
REST-PAY-AMNT-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	6.12.3	
REST-PAY-AMNT-RES-TRANS-S-003-M	Create a new amount reservation transaction for an end-user – POST (XML or JSON)	6.12.5	
REST-PAY-AMNT-RES-TRANS-S-004-O	Create a new amount reservation transaction for an end-user – POST (application/x-www-form-urlencoded)	C.3	

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

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-RES-TRANS-S-001-M	Support for access to individual completed and pending amount reservation transaction	6.13	
REST-PAY-IND-AMNT-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	6.13.3	
REST-PAY-IND-AMNT-RES-TRANS-S-003-M	Update an amount reservation transaction for an end-user – POST (XML or JSON)	6.13.5	
REST-PAY-IND-AMNT-RES-TRANS-S-004-O	Update an amount reservation transaction for an end-user (reserve an additional amount) – POST (application/x-www-form-urlencoded)	C.4	

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-RES-TRANS-S-005-O	Update an amount reservation transaction for an end-user (charge to a reservation) – POST (application/x-www-form-urlencoded)	C.5	
REST-PAY-IND-AMNT-RES-TRANS-S-006-O	Update an amount reservation transaction for an end-user (release a reservation) – POST (application/x-www-form-urlencoded)	C.6	

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

Item	Function	Reference	Requirement
REST-PAY-VOL-RES-TRANS-S-001-O	Support for access to all volume reservation transactions	6.14	REST-PAY-VOL-RES-TRANS-S-003-O
REST-PAY-VOL-RES-TRANS-S-002-O	Obtain all transactions for an end-user - GET	6.14.3	
REST-PAY-VOL-RES-TRANS-S-003-O	Create a new volume reservation transaction for an end-user - POST	6.14.5	

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

Item	Function	Reference	Requirement
REST-PAY-VOL-IND-RES-TRANS-S-001-O	Support for access to individual volume reservation transaction	6.15	REST-PAY-VOL-IND-RES-TRANS-S-003-O
REST-PAY-VOL-IND-RES-TRANS-S-002-O	Obtain individual transaction for an end-user - GET	6.15.3	
REST-PAY-VOL-IND-RES-TRANS-S-003-O	Update a volume reservation transaction for an end-user - POST	6.15.5	

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

Item	Function	Reference	Requirement
REST-PAY-IND-AMNT-VOL-S-001-O	Support for access to individual payment information based on volume reservation transaction	6.16	REST-PAY-IND-AMNT-VOL-S-002-O
REST-PAY-IND-AMNT-VOL-S-002-O	Obtain individual transaction for an end-user - GET	6.16.3	

B.1.17   SCR for REST.Payment.Amount.Converted Server

Item	Function	Reference	Requirement
REST-PAY-AMNT-CONV-S-001-O	Support for access to amount converted from given volume	6.17	REST-PAY-AMNT-CONV-S-002-O
REST-PAY-AMNT-CONV-S-002-O	Return the amount resulting from converting the given volume for an end user - GET	6.17.3	

B.1.18   SCR for REST.Payment.Async.Transaction Server

Item	Function	Reference	Requirement
REST-PAY-ASYNC-TRANS-S-001-O	Support for asynchronous payment transactions (i.e. server support of "Processing" status)	5.2.3.1	REST-PAY-ASYNC-TRANS-S-002-O AND REST-PAY-ASYNC-TRANS-S-003-O
REST-PAY-ASYNC-TRANS-S-002-O	Support for "Processing" transaction status in response to creating a transaction - POST	6.2.5, 6.3.5, 6.6.5, 6.7.5, 6.12.5, 6.13.5, 6.14.5, 6.15.5, C.1, C.2, C.3, C.4, C.5, C.6 6.18.5	
REST-PAY-ASYNC-TRANS-S-003-O	Support for retrieving a transaction status – GET	6.5.3, 6.8.3, 6.9.3, 6.10.3, 6.11.3, 6.13.3, 6.15.3, 6.16.3	
REST-PAY-ASYNC-TRANS-S-004-O	Notify application about final transaction status - POST	6.18.5	

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

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

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

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

- Charging an amount to an end user's account
- Refunding an amount to an end user' account
- Reserving a charge for an end user's account
- Adding a charge to 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, see section 6.2.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
transactionOperationStatus	xsd:string	No	Charged (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings and description)
description	xsd:string	No	Description text to be used for information and billing text
currency	xsd:string	Yes	Currency identifier as defined in [ISO4217]
amount	xsd:decimal	Yes	Amount to be charged/refunded/reserved. The amount to be charged/refunded/reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.

code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a 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 duplicate transaction creation 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.
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction.
			For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.2 and 6.2.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML.
			Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant

			if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies. How the consent is established is out of scope.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:string	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.1.1   Example 1, using 'tel' URI   (Informative)

C.1.1.1   Request

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


endUserId=tel%3A%2B19585550100&
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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>

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

```
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountCharged>10</totalAmountCharged>
  <chargingMetaData>
   <onBehalfOf>Example Games Inc</onBehalfOf>
   <purchaseCategoryCode>Game</purchaseCategoryCode>
   <channel>WAP</channel>
   <taxAmount>0</taxAmount>
  </chargingMetaData>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

C.1.2   Example 2, asynchronous transaction returning "Processing" status   (Informative)

C.1.2.1   Request

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


endUserId=tel%3A%2B19585550100&
transactionOperationStatus=Charged&
description=Test%20amount%20transaction%20%22Charged%22&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
clientCorrelator=54321&
notifyURL=http://application.example.com/notifications/PaymentTransactionFinalResult&
callbackData=12345&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.1.2.2   Response

```
HTTP/1.1 202 Accepted
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
 <!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
```

```
<endUserId>tel:+19585550100</endUserId>
<paymentAmount>
  <chargingInformation>
    <description>Test amount transaction "Processing"</description>
    <currency>USD</currency>
    <amount>10</amount>
    <code>TEST-012345</code>
  </chargingInformation>
</paymentAmount>
<transactionOperationStatus>Processing</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
<clientCorrelator>54321</clientCorrelator>
<callbackReference>
    <notifyURL>http://application.example.com/notifications/PaymentTransactionFinalResult</notifyURL>
    <callbackData>12345</callbackData>
</callbackReference>
<resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

C.1.3   Example 3, using 'acr' URI   (Informative)

C.1.3.1   Request

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


endUserId=acr%3Apseudonym123&
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.1.3.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT CHARGE TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
```

```
<endUserId>acr:pseudonym123</endUserId>
<paymentAmount>
 <chargingInformation>
  <description>Test amount transaction "Charged" </description>
  <currency>USD</currency>
  <amount>10</amount>
  <code>TEST-012345</code>
 </chargingInformation>
 <totalAmountCharged>10</totalAmountCharged>
 <chargingMetaData>
  <onBehalfOf>Example Games Inc</onBehalfOf>
  <purchaseCategoryCode>Game</purchaseCategoryCode>
  <channel>WAP</channel>
  <taxAmount>0</taxAmount>
 </chargingMetaData>
</paymentAmount>
<transactionOperationStatus>Charged</transactionOperationStatus>
<referenceCode>REF-12345</referenceCode>
<serverReferenceCode>ABC-123</serverReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123</resourceURL>
</payment:amountTransaction>
```

C.2   Refund an Amount

This operation is used to refund a currency amount to an end-user account, see section 6.2.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
transactionOperationStatus	TransactionOperationStatus	No	Refunded (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings and description)
description	xsd:string	No	Description text to be used for information and billing text
currency	xsd:string	Yes	Currency identifier as defined in [ISO4217]
amount	xsd:decimal	Yes	Amount to be charged/refunded/reserved. The amount to be charged/refunded/reserved

			appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
originalServerReferenceCode	xsd:string	Yes	This can be used to reconcile a refund request with the original charge that is intended to be refunded. In case the server included a serverReferenceCode in the response to a charge request, then any subsequent client request to refund that charge SHOULD include that serverReferenceCode value in an originalServerReferenceCode field. If the client omits it from the refund request then the server MAY throw a policy exception.
clientCorrelator	xsd:string	Yes	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 duplicate transaction creation 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.
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction. For the use of Client-side Notification URLs and Server-side Notification URLs in this

			parameter, see sections 6.2 and 6.2.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML. Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:string	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.2.1   Example   (Informative)

C.2.1.1   Request

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

endUserId=tel%3A%2B19585550100&
transactionOperationStatus=Refunded&
description=Test%20amount%20transaction%20%22Refunded%22&
currency=USD&
amount=10&
code=TEST-012345&
referenceCode=REF-12345&
originalServerReferenceCode=ABC123&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.2.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans124
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT REFUND TRANSACTION -->
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount transaction "Refunded" </description>
   <currency>USD</currency>
   <amount>10</amount>
   <code>TEST-012345</code>
  </chargingInformation>
  <totalAmountRefunded>10</totalAmountRefunded>
  <chargingMetaData>
   <onBehalfOf>Example Games Inc</onBehalfOf>
   <purchaseCategoryCode>Game</purchaseCategoryCode>
   <channel>WAP</channel>
   <taxAmount>0</taxAmount>
  </chargingMetaData>
 </paymentAmount>
 <transactionOperationStatus>Refunded</transactionOperationStatus>
 <referenceCode>REF-12345</referenceCode>
<serverReferenceCode>DEF-123</serverReferenceCode>
<originalServerReferenceCode>ABC-123</originalServerReferenceCode>
 <clientCorrelator>54321</clientCorrelator>
```

```
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans124</resourceURL>
</payment:amountTransaction>
```

C.3 Reserve an Amount

This operation is used to reserve a currency amount against an end-user account see section 6.12.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	No	Unique identifier for the end user's account (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
transactionOperationStatus	TransactionOperationStatus	No	Reserved (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings and description)
description	xsd:string	No	Description text to be used for information and billing text
currency	xsd:string	Yes	Currency identifier as defined in [ISO4217]
amount	xsd:decimal	Yes	Amount to be charged/refunded/reserved. The amount to be charged/refunded/reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceSequence	xsd:string	No	Sequential number generated by client application for every transaction state change. 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).
			Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate transaction creation 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.
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction. For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.12 and 6.12.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML.

			Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:decimal	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:string	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:anyURI	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.3.1   Example   (Informative)

C.3.1.1   Request

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

endUserId=tel%3A%2B19585550100&
transactionOperationStatus=Reserved&
description= Test%20amount%20reservation%20transaction%20%22Reserved%22&
currency=USD&
amount=10&
```

```
referenceCode=TEST-012345&
referenceSequence=1&
clientCorrelator=54321&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.3.1.2   Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction
 xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <endUserId>tel:+19585550100</endUserId>
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Reserved"</description>
   <currency>USD</currency>
   <amount>10</amount>
  </chargingInformation>
  <totalAmountCharged>0</totalAmountCharged>
  <amountReserved>10</amountReserved>
  <chargingMetaData>
   <onBehalfOf>Example Games Inc</onBehalfOf>
   <purchaseCategoryCode>Game</purchaseCategoryCode>
   <channel>WAP</channel>
   <taxAmount>0</taxAmount>
  </chargingMetaData>
 </paymentAmount>
 <transactionOperationStatus>Reserved</transactionOperationStatus>
 <referenceSequence>1</referenceSequence>
 <referenceCode>TEST-012345</referenceCode>
 <clientCorrelator>54321</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

C.4 Reserve an Additional Amount

This operation is used to add a currency amount to an existing reservation, see section 6.13.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	Yes	Unique identifier for the end user's account (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
transactionOperationStatus	TransactionOperationStatus	No	e.g. Charged, Reserved, etc (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings)
description	xsd:string	No	Description text to be used for information and billing text
currency	xsd:string	Yes	Currency identifier as defined in [ISO4217]
amount	xsd:decimal	Yes	Amount to be charged/refunded/reserved. The amount to be charged/refunded/reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes

referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change.
			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).
			Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction.
			For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.13 and 6.13.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.

notificationFormat	common:NotificationFormat	Yes	Default: XML. Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceId	xsd:anyURI	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:anyURI	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.4.1   Example   (Informative)

C.4.1.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

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

C.4.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- PENDING AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
  <paymentAmount>
    <chargingInformation>
      <description>Test amount reservation transaction "Reserved"</description>
      <amount>5</amount>
    </chargingInformation>
    <totalAmountCharged>0</totalAmountCharged>
    <amountReserved>15</amountReserved>
  </paymentAmount>
  <transactionOperationStatus>Reserved</transactionOperationStatus>
  <referenceSequence>2</referenceSequence>
  <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

C.5   Charge to a Reservation

This operation is used to charge against an existing reservation, see section 6.13.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
endUserId	xsd:anyURI	Yes	Unique identifier for the end user's account (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If endUserId is also part of the request URL, the two MUST have the same value.
transactionOperationStatus	TransactionOperationStatus	No	Charged (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings and description)
description	xsd:string	No	Description text to be used for information and billing text
currency	xsd:string	Yes	Currency identifier as defined in [ISO4217]
amount	xsd:decimal	Yes	Amount to be charged/refunded/reserved. The amount to be charged/refunded/reserved appears either directly in the amount-field or as code in the code-field. If both these two fields are missing or empty a service exception (SVC0007) will be thrown.
code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceCode	xsd:string	No	Textual information to uniquely identify the request, for example, in the case of disputes
referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change. 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). Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction. For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.13 and 6.13.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.

notificationFormat	common:Notificat ionFormat	Yes	Default: XML. Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this charge applies.
serviceId	xsd:anyURI	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:anyURI	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.5.1   Example   (Informative)

C.5.1.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

transactionOperationStatus=Charged&
description=Test%20amount%20reservation%20transaction%20in%22Charged%22&
amount=5&
referenceCode=REF-12345&
referenceSequence=3&
onBehalfOf=Example%20Games%20Inc&
purchaseCategoryCode=Game&
channel=WAP&
taxAmount=0
```

C.5.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <paymentAmount>
  <chargingInformation>
   <description>Test amount reservation transaction "Charged" </description>
   <amount>5</amount>
  </chargingInformation>
  <totalAmountCharged>5</totalAmountCharged>
  <amountReserved>10</amountReserved>
  <chargingMetaData>
   <onBehalfOf>Example Games Inc</onBehalfOf>
   <purchaseCategoryCode>Game</purchaseCategoryCode>
   <channel>WAP</channel>
   <taxAmount>0</taxAmount>
  </chargingMetaData>
 </paymentAmount>
 <transactionOperationStatus>Charged</transactionOperationStatus>
 <referenceSequence>3</referenceSequence>
 <referenceCode>REF-12345</referenceCode>
 <clientCorrelator>54321</clientCorrelator>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

C.6   Release a Reservation

This operation is used to return the funds in an existing reservation, see section 6.13.5.

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

The request parameters are as follows:

Name	Type/Values	Optional	Description
transactionOperationStatus	TransactionOperationStatus	No	Released (see TransactionOperationStatus enumeration, section 5.2.3.1 for allowed strings and description)
description	xsd:string	No	Description text to be used for information and billing text
code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied
referenceSequence	xsd:int	No	Sequential number generated by client application for every transaction state change. 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).
			Example of a sequence of usable referenceSequence values: use 1 to reserve amount X, then 2 to reserve additional amount Y, then 3 to charge reserved amount, etc.
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, for example, in the case of disputes
notifyURL	xsd:anyURI	Yes	URL used by the server to notify the application about completion of a transaction.
			For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.13 and 6.13.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.
notificationFormat	common:NotificationFormat	Yes	Default: XML.
			Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
onBehalfOf	xsd:string	Yes	String parameter to allow aggregator or acquiring partners to specify who the payment is really by.
purchaseCategoryCode	xsd:string	Yes	A category defining the type of service, product or media being purchased.
channel	xsd:string	Yes	The channel over which the requester is interacting with the merchant, based on a pre-defined list of channels (e.g. WAP, Web, SMS...) with the ability to extend the channel list as required.
taxAmount	xsd:decimal	Yes	The tax amount charged by the merchant if the charge has tax already included. This also provides an indicator to the downstream billing system.
mandateId	xsd:string	Yes	The ID representing the subscription service or consent approval for which this

			charge applies.
serviceId	xsd:string	Yes	The ID of the partner/merchant service being purchased.
productId	xsd:string	Yes	Combines with the service ID to uniquely indentify the product being purchased.

C.6.1   Example   (Informative)

C.6.1.1   Request

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

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

C.6.1.2   Response

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

<?xml version="1.0" encoding="UTF-8"?>
<!-- AMOUNT RESERVATION TRANSACTION -->
<payment:amountReservationTransaction
 xmlns:payment="urn:oma:xml:rest:netapi:payment:1">
 <paymentAmount>
  <chargingInformation>
    <description>Test amount reservation transaction "Released" </description>
  </chargingInformation>
  <totalAmountCharged>5</totalAmountCharged>
  <amountReserved>0</amountReserved>
 </paymentAmount>
 <transactionOperationStatus>Released</transactionOperationStatus>
 <referenceSequence>4</referenceSequence>
 <resourceURL>http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999</resourceURL>
</payment:amountReservationTransaction>
```

Appendix D.   JSON examples   (Informative)

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

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

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

D.1   Get all transactions (section 6.1.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
    "amountReservationTransaction": [
      {
        "clientCorrelator": "55555",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
          "amountReserved": "0",
          "chargingInformation": {
            "amount": "15",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction  \"Charged\""
          },
          "totalAmountCharged": "25"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "2",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans128",
        "serverReferenceCode": "PQR-123",
        "transactionOperationStatus": "Charged"
      },
      {
        "clientCorrelator": "55556",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
          "amountReserved": "10",
          "chargingInformation": {
            "amount": "10",
```

```
                    "code": "TEST012345",
                    "currency": "USD",
                    "description": "Test amount reservation transaction  \"Reserved\""
                 },
                "totalAmountCharged": "0"
             },
            "referenceCode": "REF-12345",
            "referenceSequence": "1",
            "resourceURL":
 "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans998",
            "transactionOperationStatus": "Reserved"
         }
    ],
    "amountSplitTransaction": {
        "clientCorrelator": "54323",
        "endUserShare": [
            {
                "endUserId": "tel:+19585550100",
                "percent": "20"
            },
            {
                "endUserId": "tel:+19585550101",
                "percent": "80"
            }
        ],
        "paymentAmount": {
            "chargingInformation": {
                "amount": "10",
                "code": "TEST-012345",
                "currency": "USD",
                "description": "Test amount transaction \"Charged\""
            },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans789",
        "serverReferenceCode": "ABC-789",
        "transactionOperationStatus": "Charged"
    },
    "amountTransaction": [
        {
            "clientCorrelator": "54321",
            "endUserId": "tel:+19585550100",
            "paymentAmount": {
            "chargingInformation": {
                "amount": "10",
                "code": "TEST-012345",
                "currency": "USD",
                "description": "Test amount transaction  \"Charged\""
             },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
```

```
      },
      {
         "clientCorrelator": "54322",
         "endUserId": "tel:+19585550100",
         "originalServerReferenceCode": "ABC-123",
         "paymentAmount": {
            "chargingInformation": {
               "amount": "10",
               "code": "TEST012345",
               "currency": "USD",
               "description": "Test amount transaction \"Refunded\""
            },
            "totalAmountRefunded": "10"
         },
         "referenceCode": "REF-12345",
         "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans456",
         "serverReferenceCode": "ABC-456",
         "transactionOperationStatus": "Refunded"
      }
   ],
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions",
   "volumeReservationTransaction": [
      {
         "clientCorrelator": "66666",
         "endUserId": "tel:+19585550100",
         "paymentVolume": {
            "billingText": "Test volume reservation transaction \"Reserved\"",
            "ratingParameter": {
               "name": "unit",
               "value": "minutes"
            },
            "totalVolumeCharged": "0",
            "volume": "10",
            "volumeReserved": "10"
         },
         "referenceCode": "REF-12345",
         "referenceSequence": "1",
         "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999",
         "transactionOperationStatus": "Reserved"
      },
      {
         "clientCorrelator": "66667",
         "endUserId": "tel:+19585550100",
         "paymentVolume": {
            "billingText": "Test volume reservation transaction \"Charged\"",
            "ratingParameter": {
               "name": "unit",
               "value": "minutes"
            },
            "totalVolumeCharged": "25",
            "volume": "10",
            "volumeReserved": "0"
         },
         "referenceCode": "REF-12345",
         "referenceSequence": "4",
```

```
              "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans127",
              "serverReferenceCode": "MNO-123",
              "transactionOperationStatus": "Charged"
          }
      ],
      "volumeSplitTransaction": {
          "clientCorrelator": "54325",
          "endUserShare": [
              {
                  "endUserId": "tel:+19585550100",
                  "percent": "20"
              },
              {
                  "endUserId": "tel:+19585550101",
                  "percent": "80"
              }
          ],
          "paymentVolume": {
              "billingText": "Test volume transaction  \"Charged\"",
              "ratingParameter": {
                  "name": "unit",
                  "value": "minutes"
              },
              "totalVolumeCharged": "10",
              "volume": "10"
          },
          "referenceCode": "REF-12345",
          "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans126",
          "serverReferenceCode": "JKL-123",
          "transactionOperationStatus": "Charged"
      },
      "volumeTransaction": [
          {
              "clientCorrelator": "55551",
              "endUserId": "tel:+19585550100",
              "paymentVolume": {
                  "billingText": "Test volume transaction \"Charged\"",
                  "ratingParameter": {
                      "name": "unit",
                      "value": "minutes"
                  },
                  "totalVolumeCharged": "10",
                  "volume": "10"
              },
              "referenceCode": "REF-12345",
              "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124",
              "serverReferenceCode": "DEF-123",
              "transactionOperationStatus": "Charged"
          },
          {
              "clientCorrelator": "54324",
              "endUserId": "tel:+19585550100",
              "originalServerReferenceCode": "DEF-123",
              "paymentVolume": {
                  "billingText": "Test volume transaction \"Refunded\"",
```

```
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeRefunded": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans125",
    "serverReferenceCode": "GHI-123",
    "transactionOperationStatus": "Refunded"
   }
 ]
}}
```

D.2   Request with invalid endUserId (section 6.1.3.2)

Request:

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

Response:

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


{"requestError": {
   "link": {
       "href": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions",
       "rel": "PaymentTransactionList"
   },
   "serviceException": {
       "messageId": "SVC0004",
       "text": "No valid addresses provided in message part %1",
       "variables": "endUserId=tel:+19585550100"
   }
}}
```

D.3   Get all amount transactions (section 6.2.3.1)

Request:

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

Response:

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


{"paymentTransactionList": {
   "amountTransaction": [
      {
         "clientCorrelator": "54321",
         "endUserId": "tel:+19585550100",
         "paymentAmount": {
            "chargingInformation": {
               "amount": "10",
               "code": "TEST-012345",
               "currency": "USD",
               "description": "Test amount transaction \"Charged\""
            },
            "totalAmountCharged": "10"
         },
         "referenceCode": "REF-12345",
         "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
         "serverReferenceCode": "ABC-123",
         "transactionOperationStatus": "Charged"
      },
      {
         "clientCorrelator": "54321",
         "endUserId": "tel:+19585550100",
         "paymentAmount": {
            "chargingInformation": {
               "amount": "10",
               "code": "TEST012345",
               "currency": "USD",
               "description": "Test amount transaction \"Charged\""
            },
            "totalAmountCharged": "10"
         },
         "referenceCode": "REF-12345",
         "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans124",
         "serverReferenceCode": "DEF-123",
         "transactionOperationStatus": "Charged"
      },
      {
         "clientCorrelator": "54330",
         "endUserId": "tel:+19585550100",
         "originalServerReferenceCode": "DEF-123",
         "paymentAmount": {
            "chargingInformation": {
               "amount": "10",
               "code": "TEST012345",
               "currency": "USD",
               "description": "Test amount transaction \"Refunded\""
            },
            "totalAmountRefunded": "10"
```

```
        },
        "referenceCode": "REF-12345",
        "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans129",
        "serverReferenceCode": "WXY-123",
        "transactionOperationStatus": "Refunded"
      }
    ],
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount"
}}
```

D.4   Create charge amount using 'tel' URI (section 6.2.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+19585550100",
    "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+19585550100",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\""
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
```

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

D.5   Create charge amount with callbackReference and "Processing" status returned (section 6.2.5.2)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/PaymentTransactionFinalResult"
   },
   "clientCorrelator": "54321",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {"chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Processing\""
   }},
   "referenceCode": "REF-12345",
   "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 202 Accepted
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/PaymentTransactionFinalResult"
   },
   "clientCorrelator": "54321",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "chargingInformation": {
         "amount": "10",
         "code": "TEST-012345",
         "currency": "USD",
         "description": "Test amount transaction \"Processing\""
      }
```

```
     },
     "referenceCode": "REF-12345",
     "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
     "serverReferenceCode": "ABC-123",
     "transactionOperationStatus": "Processing"
}}
```

D.6   Create refund amount (section 6.2.5.3)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+19585550100",
    "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans130
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+19585550100",
    "originalServerReferenceCode": "ABC-123",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Refunded\""
        },
        "totalAmountRefunded": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans130",
    "serverReferenceCode": "XYZ-123",
```

```
      "transactionOperationStatus": "Refunded"
}}
```

D.7   Client retries POST with same clientCorrelator (section 6.2.5.4)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:+19585550100",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn


{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "chargingInformation": {
         "amount": "10",
         "code": "TEST-012345",
         "currency": "USD",
         "description": "Test amount transaction \"Charged\" "
      },
      "totalAmountCharged": "10"
   },
   "referenceCode": "REF-12345",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
   "serverReferenceCode": "ABC-123",
   "transactionOperationStatus": "Charged"
}}
```

D.8   Unsuccessful charge request because of denial/refusal by back-end system (section 6.2.5.5)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:+19585550100",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

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

D.9   Create charge amount using 'acr' URI (section 6.2.5.6)

Request:

```
POST /exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "acr:pseudonym123",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123
Content-Type: application/json
Content-Length: nnnn

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "acr:pseudonym123",
   "paymentAmount": {
      "chargingInformation": {
         "amount": "10",
         "code": "TEST-012345",
         "currency": "USD",
         "description": "Test amount transaction \"Charged\""
      },
      "totalAmountCharged": "10"
   },
   "referenceCode": "REF-12345",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/acr%3Apseudonym123/transactions/amount/trans123",
   "serverReferenceCode": "ABC-123",
   "transactionOperationStatus": "Charged"
}}
```

D.10 Get all amount split transactions (section 6.3.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
   "amountSplitTransaction": [
      {
         "clientCorrelator": "55552",
         "endUserShare": [
            {
               "endUserId": "tel:+19585550100",
               "percent": "30"
            },
            {
```

```
              "endUserId": "tel:+19585550101",
              "percent": "70"
            }
          ],
          "paymentAmount": {
            "chargingInformation": {
              "amount": "20",
              "code": "TEST-012345",
              "currency": "USD",
              "description": "Test amount transaction \"Charged\""
            },
            "totalAmountCharged": "20"
          },
          "referenceCode": "REF-12345",
          "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123",
          "serverReferenceCode": "ABC-123",
          "transactionOperationStatus": "Charged"
        },
        {
          "clientCorrelator": "55553",
          "endUserShare": [
            {
              "endUserId": "tel:+19585550100",
              "percent": "20"
            },
            {
              "endUserId": "tel:+19585550101",
              "percent": "80"
            }
          ],
          "paymentAmount": {
            "chargingInformation": {
              "amount": "10",
              "code": "TEST-012346",
              "currency": "USD",
              "description": "Test amount transaction \"Charged\" "
            },
            "totalAmountCharged": "10"
          },
          "referenceCode": "REF-12346",
          "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans124",
          "serverReferenceCode": "DEF-123",
          "transactionOperationStatus": "Charged"
        }
      ],
      "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit"
}}
```

D.11 Create split charge amount (section 6.3.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountSplitTransaction": {
    "clientCorrelator": "54431",
    "endUserShare": [
        {
            "endUserId": "tel:+19585550100",
            "percent": "30"
        },
        {
            "endUserId": "tel:+19585550101",
            "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123
Content-Type: application/json
Content-Length: nnnn

{"amountSplitTransaction": {
    "clientCorrelator": "54431",
    "endUserShare": [
        {
            "endUserId": "tel:+19585550100",
            "percent": "30"
        },
        {
            "endUserId": "tel:+19585550101",
            "percent": "70"
        }
    ],
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
```

```
      "currency": "USD",
      "description": "Test amount transaction \"Charged\" "
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}
```

D.12 Get amount charge (section 6.4.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123?resFormat=JSON HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"amountTransaction": {
  "clientCorrelator": "54321",
  "endUserId": "tel:+19585550100",
  "paymentAmount": {
    "chargingInformation": {
      "amount": "10",
      "code": "TEST-012345",
      "currency": "USD",
      "description": "Test amount transaction \"Charged\" "
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amount/trans123",
  "serverReferenceCode": "ABC-123",
  "transactionOperationStatus": "Charged"
}}
```

D.13 Get amount split charge (section 6.5.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"amountSplitTransaction": {
    "clientCorrelator": "54321",
    "endUserShare": [
        {
            "endUserId": "tel:+19585550100",
            "percent": "30"
        },
        {
            "endUserId": "tel:+19585550101",
            "percent": "70"
        }
    ],
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\" "
        },
        "totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountSplit/trans123",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
}}
```

D.14 Get all volume charge and refund transactions (section 6.6.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume",
    "volumeTransaction": [
        {
            "clientCorrelator": "55551",
            "endUserId": "tel:+19585550100",
            "paymentVolume": {
                "billingText": "Test volume transaction \"Charged\" ",
                "ratingParameter": {
                    "name": "unit",
                    "value": "minutes"
                },
                "totalVolumeCharged": "10",
                "volume": "10"
            },
            "referenceCode": "REF-12345",
            "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123",
            "serverReferenceCode": "ABC-123",
            "transactionOperationStatus": "Charged"
        },
        {
            "clientCorrelator": "55552",
            "endUserId": "tel:+19585550100",
            "originalServerReferenceCode": "ABC-123",
            "paymentVolume": {
                "billingText": "Test volume transaction \"Refunded\"",
                "ratingParameter": {
                    "name": "unit",
                    "value": "minutes"
                },
                "totalVolumeRefunded": "10",
                "volume": "10"
            },
            "referenceCode": "REF-12345",
            "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124",
            "serverReferenceCode": "DEF-123",
            "transactionOperationStatus": "Refunded"
        }
    ]
}}
```

D.15 Create charge volume, returning a representation of created resource (section 6.6.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeTransaction": {
    "clientCorrelator": "55551",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123
Content-Type: application/json
Content-Length: nnnn

{"volumeTransaction": {
    "clientCorrelator": "55551",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
}}
```

D.16 Create charge volume, returning the location of created resource (section 6.6.5.2)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn


{"volumeTransaction": {
   "clientCorrelator": "55551",
   "endUserId": "tel:+19585550100",
   "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\" ",
      "ratingParameter": {
         "name": "unit",
         "value": "minutes"
      },
      "volume": "10"
   },
   "referenceCode": "REF-12345",
   "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123
Content-Type: application/json
Content-Length: nnnn

{"resourceReference": {"resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123"}}
```

D.17 Create refund volume (section 6.6.5.3)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeTransaction": {
   "clientCorrelator": "55552",
   "endUserId": "tel:+19585550100",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124
Content-Type: application/json
Content-Length: nnnn

{"volumeTransaction": {
  "clientCorrelator": "55552",
  "endUserId": "tel:+19585550100",
  "originalServerReferenceCode": "ABC-123",
  "paymentVolume": {
    "billingText": "Test volume transaction \"Refunded\"",
    "ratingParameter": {
      "name": "unit",
      "value": "minutes"
    },
    "totalVolumeRefunded": "10",
    "volume": "10"
  },
  "referenceCode": "REF-12345",
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans124",
  "serverReferenceCode": "DEF-123",
  "transactionOperationStatus": "Refunded"
}}
```

D.18 Get all volume split charge transactions (section 6.7.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit",
  "volumeSplitTransaction": {
```

```
      "clientCorrelator": "55553",
      "endUserShare": [
         {
            "endUserId": "tel:+19585550100",
            "percent": "20"
         },
         {
            "endUserId": "tel:+19585550101",
            "percent": "80"
         }
      ],
      "paymentVolume": {
         "billingText": "Test volume transaction \"Charged\"",
         "ratingParameter": {
            "name": "unit",
            "value": "minutes"
         },
         "totalVolumeCharged": "10",
         "volume": "10"
      },
      "referenceCode": "REF-12345",
      "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123",
      "serverReferenceCode": "ABC-123",
      "transactionOperationStatus": "Charged"
   }
}}
```

D.19 Create volume split charge (section 6.7.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeSplitTransaction": {
   "clientCorrelator": "55553",
   "endUserShare": [
      {
         "endUserId": "tel:+19585550100",
         "percent": "20"
      },
      {
         "endUserId": "tel:+19585550101",
         "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123
Content-Type: application/json
Content-Length: nnnn

{"volumeSplitTransaction": {
   "clientCorrelator": "55553",
   "endUserShare": [
      {
         "endUserId": "tel:+19585550100",
         "percent": "20"
      },
      {
         "endUserId": "tel:+19585550101",
         "percent": "80"
      }
   ],
   "paymentVolume": {
      "billingText": "Test volume transaction \"Charged\" ",
      "ratingParameter": {
         "name": "unit",
         "value": "minutes"
      },
      "volume": "10"
   },
   "referenceCode": "REF-12345",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123",
   "serverReferenceCode": "ABC-123",
   "transactionOperationStatus": "Charged"
}}
```

D.20 Get volume charge (section 6.8.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

```
{"volumeTransaction": {
    "clientCorrelator": "55555",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans123",
    "serverReferenceCode": "ABC-123",
    "transactionOperationStatus": "Charged"
}}
```

D.21 Get volume split charge (section 6.9.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"volumeSplitTransaction": {
    "clientCorrelator": "55556",
    "endUserShare": [
        {
            "endUserId": "tel:+19585550100",
            "percent": "20"
        },
        {
            "endUserId": "tel:+19585550101",
            "percent": "80"
        }
    ],
    "paymentVolume": {
        "billingText": "Test volume transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10"
    },
    "referenceCode": "REF-12345",
```

```
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans123",
   "serverReferenceCode": "ABC-123",
   "transactionOperationStatus": "Charged"
}}
```

D.22 Get amount for volume charge (section 6.10.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volume/trans999/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"paymentAmount": {"chargingInformation": {
   "amount": "10",
   "currency": "USD",
   "description": "10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+19585550100"
}}}
```

D.23 Get amount for volume split charge (section 6.11.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeSplit/trans999/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

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

D.24 Get all amount reservation transactions (section 6.12.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
    "amountReservationTransaction": [
      {
        "clientCorrelator": "55555",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
          "amountReserved": "0",
          "chargingInformation": {
            "amount": "15",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Charged\" "
          },
          "totalAmountCharged": "25"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "2",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans123",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
      },
      {
        "clientCorrelator": "55556",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
          "amountReserved": "10",
          "chargingInformation": {
            "amount": "10",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Reserved\""
          },
          "totalAmountCharged": "0"
        },
        "referenceSequence": "1",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
        "transactionOperationStatus": "Reserved"
      },
```

```
    {
        "clientCorrelator": "55557",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
            "amountReserved": "15",
            "chargingInformation": {
                "amount": "15",
                "code": "TEST012345",
                "currency": "USD",
                "description": "Test amount reservation transaction \"Reserved\""
            },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "1",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans998",
        "transactionOperationStatus": "Reserved"
    },
    {
        "clientCorrelator": "55557",
        "endUserId": "tel:+19585550100",
        "paymentAmount": {
            "amountReserved": "25",
            "chargingInformation": {
                "amount": "10",
                "code": "TEST012345",
                "currency": "USD",
                "description": "Test amount reservation transaction \"Reserved\""
            },
            "totalAmountCharged": "10"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "3",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans997",
        "transactionOperationStatus": "Reserved"
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation"
}}
```

D.25 Create reserve amount (section 6.12.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
    "clientCorrelator": "55555",
    "endUserId": "tel:+19585550100",
```

```
"paymentAmount": {"chargingInformation": {
   "amount": "10",
   "code": "TEST-012345",
   "currency": "USD",
   "description": "Test amount reservation transaction \"Reserved\""
}},
"referenceSequence": "1",
"transactionOperationStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "clientCorrelator": "55555",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "amountReserved": "10",
      "chargingInformation": {
         "amount": "10",
         "code": "TEST-012345",
         "currency": "USD",
         "description": "Test amount reservation transaction \"Reserved\""
      },
      "totalAmountCharged": "0"
   },
   "referenceSequence": "1",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
   "transactionOperationStatus": "Reserved"
}}
```

D.26 Get amount reservation (section 6.13.3.1)

Request:

```
GET /exampleAPI/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"amountReservationTransaction": {
   "clientCorrelator": "55555",
   "endUserId": "tel:+19585550100",
```

```
   "paymentAmount": {
      "amountReserved": "10",
      "chargingInformation": {
         "amount": "10",
         "code": "TEST012345",
         "currency": "USD",
         "description": "Test amount reservation transaction \"Reserved\""
      },
      "totalAmountCharged": "0"
   },
   "referenceSequence": "1",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
   "transactionOperationStatus": "Reserved"
}}
```

D.27 Charge amount for amount reservation (section 6.13.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "endUserId": "tel:+19585550100",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "clientCorrelator": "55555",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "amountReserved": "0",
      "chargingInformation": {
         "amount": "10",
         "code": "TEST012345",
         "currency": "USD",
         "description": "Test amount reservation transaction \"Charged\" "
```

```
    },
    "totalAmountCharged": "10"
  },
  "referenceCode": "REF-12345",
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Charged"
}}
```

D.28 Repeat charge request with same referenceSequence for an amount reservation (section 6.13.5.2)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "endUserId": "tel:+19585550100",
   "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "clientCorrelator": "55555",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "amountReserved": "0",
      "chargingInformation": {
         "amount": "10",
         "code": "TEST012345",
         "currency": "USD",
         "description": "Test amount reservation transaction \"Charged\" "
      },
      "totalAmountCharged": "10"
   },
   "referenceCode": "REF-12345",
```

```
   "referenceSequence": "2",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
   "transactionOperationStatus": "Charged"
}}
```

D.29 Release amount reservation (section 6.13.5.3)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
   "endUserId": "tel:+19585550100",
   "paymentAmount": {"chargingInformation": {
      "code": "TEST012345",
      "description": "Test amount reservation transaction \"Released\""
   }},
   "referenceSequence": "2",
   "transactionOperationStatus": "Released"
}}
```

Response:

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

{"amountReservationTransaction": {
   "clientCorrelator": "55556",
   "endUserId": "tel:+19585550100",
   "paymentAmount": {
      "amountReserved": "0",
      "chargingInformation": {
         "amount": "10",
         "code": "TEST012345",
         "currency": "USD",
         "description": "Test amount reservation transaction \"Released\""
      },
      "totalAmountCharged": "0"
   },
   "referenceSequence": "2",
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
   "transactionOperationStatus": "Released"
}}
```

D.30 Charge partial amount for amount reservation (section 6.13.5.4)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
    "endUserId": "tel:+19585550100",
    "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
    "clientCorrelator": "55557",
    "endUserId": "tel:+19585550100",
    "paymentAmount": {
        "amountReserved": "5",
        "chargingInformation": {
            "amount": "5",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Charged\" "
        },
        "totalAmountCharged": "5"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
    "transactionOperationStatus": "Charged"
}}
```

D.31 Release remaining amount reservation (section 6.13.5.5)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
    "endUserId": "tel:+19585550100",
    "paymentAmount": {"chargingInformation": {
        "code": "TEST012345",
        "description": "Test amount reservation transaction \"Released\""
    }},
    "referenceSequence": "3",
    "transactionOperationStatus": "Released"
}}
```

Response:

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

{"amountReservationTransaction": {
    "clientCorrelator": "55558",
    "endUserId": "tel:+19585550100",
    "paymentAmount": {
        "amountReserved": "0",
        "chargingInformation": {
            "amount": "5",
            "code": "TEST012345",
            "currency": "USD",
            "description": "Test amount reservation transaction \"Released\""
        },
        "totalAmountCharged": "5"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "3",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
    "transactionOperationStatus": "Released"
}}
```

D.32 Reserve additional amount for amount reservation (section 6.13.5.6)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
```

```
  "endUserId": "tel:+19585550100",
  "paymentAmount": {"chargingInformation": {
    "amount": "5",
    "code": "TEST012345",
    "currency": "USD",
    "description": "Test amount reservation transaction \"Reserved\""
  }},
  "referenceSequence": "2",
  "transactionOperationStatus": "Reserved"
}}
```

Response:

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

{"amountReservationTransaction": {
  "clientCorrelator": "55559",
  "endUserId": "tel:+19585550100",
  "paymentAmount": {
    "amountReserved": "15",
    "chargingInformation": {
      "amount": "10",
      "code": "TEST012345",
      "currency": "USD",
      "description": "Test amount reservation transaction \"Reserved\""
    },
    "totalAmountCharged": "0"
  },
  "referenceSequence": "2",
  "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
  "transactionOperationStatus": "Reserved"
}}
```

D.33 Unsuccessful charge amount for amount reservation because of denial/refusal by back-end system (section 6.13.5.7)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"amountReservationTransaction": {
  "endUserId": "tel:+19585550100",
  "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
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"requestError": {
   "link": {
      "href": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/amountReservation/trans999",
      "rel": "AmountReservationTransaction"
   },
   "serviceException": {
      "messageId": "SVC0270",
      "text": "Charging operation failed, the charge was not applied."
   }
}}
```

D.34 Get all volume reservation transactions (section 6.14.3.1)

Request:

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

Response:

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

{"paymentTransactionList": {
   "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation",
   "volumeReservationTransaction": [
      {
         "clientCorrelator": "66666",
         "endUserId": "tel:+19585550100",
         "paymentVolume": {
            "billingText": "Test volume reservation transaction \"Reserved\"",
            "ratingParameter": {
               "name": "unit",
               "value": "minutes"
            },
```

```
          "totalVolumeCharged": "0",
          "volume": "10",
          "volumeReserved": "10"
        },
        "referenceSequence": "1",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999",
        "transactionOperationStatus": "Reserved"
      },
      {
        "clientCorrelator": "66667",
        "endUserId": "tel:+19585550100",
        "paymentVolume": {
          "billingText": "Test volume reservation transaction \"Reserved\"",
          "ratingParameter": {
            "name": "unit",
            "value": "minutes"
          },
          "totalVolumeCharged": "15",
          "volume": "10",
          "volumeReserved": "10"
        },
        "referenceCode": "REF-12345",
        "referenceSequence": "3",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans998",
        "transactionOperationStatus": "Reserved"
      },
      {
        "clientCorrelator": "66668",
        "endUserId": "tel:+19585550100",
        "paymentVolume": {
          "billingText": "Test volume reservation transaction \"Charged\" ",
          "ratingParameter": {
            "name": "unit",
            "value": "minutes"
          },
          "totalVolumeCharged": "25",
          "volume": "10",
          "volumeReserved": "0"
        },
        "referenceCode": "REF-22345",
        "referenceSequence": "4",
        "resourceURL":
"http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans123",
        "serverReferenceCode": "ABC-123",
        "transactionOperationStatus": "Charged"
      }
    ]
}}
```

D.35 Create reserve volume (section 6.14.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
    "clientCorrelator": "66666",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume reservation transaction \"Reserved\"",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "volume": "10"
    },
    "referenceSequence": "1",
    "transactionOperationStatus": "Reserved"
}}
```

Response:

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
    "clientCorrelator": "66666",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume reservation transaction \"Reserved\"",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "0",
        "volume": "10",
        "volumeReserved": "10"
    },
    "referenceSequence": "1",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999",
    "transactionOperationStatus": "Reserved"
}}
```

D.36 Create reserve volume with invalid endUserID (section 6.14.5.2)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
   "clientCorrelator": "66667",
   "endUserId": "tel:+19585550100",
   "paymentVolume": {
      "billingText": "Test volume reservation transaction \"Reserved\"",
      "ratingParameter": {
         "name": "unit",
         "value": "minutes"
      },
      "volume": "10"
   },
   "referenceSequence": "1",
   "transactionOperationStatus": "Reserved"
}}
```

Response:

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

{"requestError": {"serviceException": {
   "messageId": "SVC0004",
   "text": "No valid addresses provided in message part %1",
   "variables": "endUserId=tel:+19585550100"
}}}
```

D.37 Get volume reservation (section 6.15.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"volumeReservationTransaction": {
    "clientCorrelator": "66666",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume reservation transaction \"Reserved\"",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "0",
        "volume": "10",
        "volumeReserved": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "1",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999",
    "transactionOperationStatus": "Reserved"
}}
```

D.38 Charge volume (section 6.15.5.1)

Request:

```
POST /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999 HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

{"volumeReservationTransaction": {
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume reservation transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "volume": "10"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "transactionOperationStatus": "Charged"
}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

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

{"volumeReservationTransaction": {
    "clientCorrelator": "66666",
    "endUserId": "tel:+19585550100",
    "paymentVolume": {
        "billingText": "Test volume reservation transaction \"Charged\" ",
        "ratingParameter": {
            "name": "unit",
            "value": "minutes"
        },
        "totalVolumeCharged": "10",
        "volume": "10",
        "volumeReserved": "0"
    },
    "referenceCode": "REF-12345",
    "referenceSequence": "2",
    "resourceURL": "http://example.com/exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999",
    "transactionOperationStatus": "Charged"
}}
```

D.39 Get amount for volume reservation (section 6.16.3.1)

Request:

```
GET /exampleAPI/payment/v1/tel%3A%2B19585550100/transactions/volumeReservation/trans999/paymentAmount HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"paymentAmount": {"chargingInformation": {
    "amount": "10",
    "currency": "USD",
    "description": "10 Minutes converted to USD for transaction=trans999 of endUserId=tel:+19585550100"
}}}
```

D.40 Get amount converted from volume (section 6.17.3.1)

Request:

```
GET
/exampleAPI/payment/v1/tel%3A%2B19585550100/convertedVolume/paymentAmount?volume=100&unit="Minutes"&contract="MON
THLY-PLAN-001"&currency="USD" HTTP 1.1
Accept: application/json
Host: example.com
```

Response:

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

{"paymentAmount": {"chargingInformation": {
    "amount": "10",
    "currency": "USD",
    "description": "100 Minutes converted to USD for endUserId=tel:+19585550100 according to contract=\"MONTHLY-PLAN-001\""
}}}
```

D.41 Client notification about payment transaction amount (section 6.18.5.1)

Request:

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


{"paymentTransactionNotification": {"amountTransaction": {
    "clientCorrelator": "54321",
    "endUserId": "tel:+19585550100",
    "paymentAmount": {
        "chargingInformation": {
            "amount": "10",
            "code": "TEST-012345",
            "currency": "USD",
            "description": "Test amount transaction \"Charged\""
        },
        " totalAmountCharged": "10"
    },
    "referenceCode": "REF-12345",
    "transactionOperationStatus": "Charged"
}}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:52:00 GMT
```

Appendix E.   Parlay X operations mapping   (Informative)

The table below illustrates the mapping between REST resources/operations and Parlay X [3GPP 29.199-6] equivalent operations.

REST Resource	REST method	REST section reference	Parlay X equivalent operations
All amount charge and refund transactions for an end user	POST	6.2.5	chargeAmount and refundAmount
All amount split charge transactions for an end user	POST	6.3.5	chargeSplitAmount
All volume charge and refund transactions for an end user	POST	6.6.5	chargeVolume and refundVolume
All volume split charge transactions for an end user	POST	6.7.5	chargeSplitVolume
All amount reservation transactions for an end user	POST	6.12.5	reserveAmount
Individual amount reservation transaction for an end user	POST	6.13.5	reserveAdditionalAmount, chargeReservation and releaseReservation
All volume reservation transactions for an end user	POST	6.14.5	reserveVolume
Individual volume reservation transaction for an end user	POST	6.15.5	reserveAdditionalVolume, chargeReservation and releaseReservation
Amount converted from given volume	GET	6.17.3	getAmount

Table 1: Parlay X operations mapping

Appendix F.   Light-weight resources   (Informative)

As this version of the specification does not define any light-weight resources, this appendix is empty.

Appendix G.   Authorization aspects   (Normative)

This appendix specifies how to use the RESTful Payment API in combination with some authorization frameworks.

G.1   Use with OMA Authorization Framework for Network APIs

The RESTful Payment API MAY support the authorization framework defined in [Autho4API_10].

A RESTful Payment API supporting [Autho4API_10]:

- SHALL conform to section D.1 of [REST_NetAPI_Common];

- SHALL conform to this section G.1.

G.1.1   Scope values

G.1.1.1   Definitions

In compliance with [Autho4API_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Payment API:

- SHALL support the scope values defined in the table below;

- MAY support scope values not defined in this specification.

Scope value	Description	For one-time access token
oma_rest_payment.all_{apiVersion}	Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the "apiVersion" URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.	Yes
oma_rest_payment.chg	Provide access to all defined operations on amount/split-amount charge/refund and volume/split-volume charge/refund transactions.	Yes
oma_rest_payment.res	Provide access to all defined operations on amount and volume reservation transactions.	Yes

Table 2: Scope values for RESTful Payment API

G.1.1.2   Downscoping

In the case where the client requests authorization for "oma_rest_payment.all_{apiVersion}" scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- "oma_rest_payment.chg"

- "oma_rest_payment.res"

G.1.1.3   Mapping with resources and methods

Tables in this section specify how the scope values defined in section G.1.1.1 for the RESTful Payment API map to the REST resources and methods of this API. In these tables, the root "oma_rest_payment." of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/payment/{apiVersion}	Section refe-rence	HTTP verbs			
			GET	PUT	POST	DELETE
All payment transactions for an end user	/{endUserId}/transactions	6.1	**all_{apiVersion}** or **chg** or **res**	n/a	n/a	n/a

Table 3: Required scope values for: Read all Payment Transactions

Resource	URL Base URL: http://{serverRoot}/payment/{apiVersion}	Section refe-rence	HTTP verbs			
			GET	PUT	POST	DELETE
All amount charge and refund transactions for an end user	/{endUserId}/transactions/amount	6.2	**all_{apiVersion}** or **chg**	n/a	**all_{apiVersion}** or **chg**	n/a
All amount split charge transactions for an end user	/{endUserId}/transactions/amountSplit	6.3	**all_{apiVersion}** or **chg**	n/a	**all_{apiVersion}** or **chg**	n/a
Individual amount charge or refund transaction for an end user	/{endUserId}/transactions/amount/{transactionId}	6.4	**all_{apiVersion}** or **chg**	n/a	n/a	n/a

Individual amount split charge transaction for an end user	/{endUserId}/transactions/amountSplit/{transactionId}	6.5	**all_{apiVersion}** or **chg**	n/a	n/a	n/a
All volume charge and refund transactions for an end user	/{endUserId}/transactions/volume	6.6	**all_{apiVersion}** or **chg**	n/a	**all_{apiVersion}** or **chg**	n/a
All volume split charge transactions for an end user	/{endUserId}/transactions/volumeSplit	6.7	**all_{apiVersion}** or **chg**	n/a	**all_{apiVersion}** or **chg**	n/a
Individual volume charge or refund transaction for an end user	/{endUserId}/transactions/volume/{transactionId}	6.8	**all_{apiVersion}** or **chg**	n/a	n/a	n/a
Individual volume split charge transaction for an end user	/{endUserId}/transactions/volumeSplit/{transactionId}	6.9	**all_{apiVersion}** or **chg**	n/a	n/a	n/a

Individual amount for volume charge or refund transaction for an end user	/{endUserId}/transactions/volume/{transactionId}/paymentAmount	6.10	**all_{apiVersion}** or **chg**	n/a	n/a	n/a
Individual amount for volume split charge transaction for an end user	/{endUserId}/transactions/volumeSplit/{transactionId}/paymentAmount	6.11	**all_{apiVersion}** or **chg**	n/a	n/a	n/a

Table 4: Required scope values for: amount and volume charge transactions

Resource	URL Base URL: http://{serverRoot}/payment/{apiVersion}	Section refe-rence	HTTP verbs			
			GET	PUT	POST	DELETE
All amount reservation transactions for an end user	/{endUserId}/transactions/amountReservation	6.12	**all_{apiVersion}** or **res**	n/a	**all_{apiVersion}** or **res**	n/a
Individual amount reservation transaction for an end user	/{endUserId}/transactions/amountReservation/{transactionId}	6.13	**all_{apiVersion}** or **res**	n/a	**all_{apiVersion}** or **res**	n/a
All volume reservation transactions for an end user	/{endUserId}/transactions/volumeReservation	6.14	**all_{apiVersion}** or **res**	n/a	**all_{apiVersion}** or **res**	n/a

Individual volume reservation transaction for an end user	/{endUserId}/transactions/volumeReservation/{transactionId}	6.15	**all_{apiVersion}** or **res**	n/a	**all_{apiVersion}** or **res**	n/a
Individual amount for volume reservation transaction for an end user	/{endUserId}/transactions/volumeReservation/{transactionId}/paymentVolume	6.16	**all_{apiVersion}** or **res**	n/a	n/a	n/a
Amount converted from given volume	/{endUserId}/convertedVolume/paymentAmount?volume={volume}&unit={Minutes|Bytes|...}&contract={contract id}&service={serviceId}&operation={operationId}	6.17	**all_{apiVersion}** or **res**	n/a	n/a	n/a

Table 5: Required scope values for: amount and volume reservation transactions

G.1.2   Use of 'acr:auth'

This section specifies the use of 'acr:auth' in place of an end user identifier in a resource URL path.

An 'acr' URI of the form 'acr:auth', where 'auth' is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

A client MAY use 'acr:auth' in a resource URL in place of the {endUserId} resource URL variable in the resource URL path, when the RESTful Payment API is used in combination with [Autho4API_10].

In the case the RESTful Payment supports [Autho4API_10], the server:

−   SHALL accept 'acr:auth' as a valid value for the resource URL variable {endUserId}.

−   SHALL conform to [REST_NetAPI_Common] section 5.8.1.1 regarding the processing of 'acr:auth'.

Appendix H.   Mapping of RESTful Payment API Reservation Transactions to the Charging Enabler   (Informative)

RESTful Payment API 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 RESTful Payment API 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 / RESTful Payment API.

RESTful Payment API contains functionality for Reservation-based payment operations (sections 6.12, 6.13, 6.14, 6.15). 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 RESTful Payment API Reservation Transactions to the Reserve Units and Debit Units operations can be achieved.

H.1   Charging Mechanism

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

The three phases of a Charging session are as follows:

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

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

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

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

H.2   Detailed Mapping of RESTful Payment API to Charging Enabler

The previous section has outlined the basic flow in Session Charging, and the different approaches in ParlayX / RESTful Payment API 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 RESTful Payment API (see section 5.2.3.1) 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 RESTful Payment API elements amountReserved and volumeReserved.

The variable totalAmountVolumeCharged corresponds to the RESTful Payment API elements totalAmountCharged and totalVolumeCharged. The variable amountVolume corresponds to the RESTful Payment API 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 RESTful Payment API.

1)   initial reserving as part of RESTful Payment API resource creation

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

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

2)   charging as RESTful Payment API resource update

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

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

3)   reserving as RESTful Payment API resource update

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

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

4)   releasing as RESTful Payment API resource update

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

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

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

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