RESTful bindings for Parlay X Web Services – Multi-media Messaging

Candidate Version 1.0 – 27 Apr 2010

Open Mobile Alliance

OMA-TS-ParlayREST_MultiMediaMessaging-V1_0-20100427-C

Contents

Figures

1. Scope

This specification defines an HTTP protocol binding for an abstract API using the REST architectural style, based on existing OMA enabler namely the Multi Media Messaging Service, as defined in [3GPP 29.199-5].

2. References

2.1 Normative References

[3GPP 23.140]	3GPP Technical Specification, "Multimedia Messaging Service (MMS); Functional description; Stage 2 (Release 6)", URL:http://www.3gpp.org/
[3GPP 29.199-5]	3GPP Technical Specification, "Open Service Access (OSA); Parlay X Web Services; Part 5: Multimedia messaging (Release 8)", URL:http://www.3gpp.org/
[OMA-IM-TS]	"Instant Messaging using SIMPLE", Open Mobile Alliance™, OMA-TS-SIMPLE_IM-V1_0-20100218-D.doc, URL: http://www.openmobilealliance.org/
[REST_TS_Common]	"RESTful bindings for Parlay X Web Services – Common", Open Mobile Alliance™, OMA-TS-ParlayREST_Common-V1_0, URL:http://www.openmobilealliance.org/
[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt
[RFC2388]	"Returning Values from Forms: multipart/form-data" L.Masinter. August, 1998. URL:http://www.ietf.org/rfc/rfc2388.txt
[RFC2616]	"Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding et. al, January 1999, URL:http://www.ietf.org/rfc/rfc2616.txt
[RFC4234]	"Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt
[RFC4627]	"The application/json Media Type for JavaScript Object Notation (JSON)", D. Crockford, July 2006, URL:http://www.ietf.org/rfc/rfc4627.txt
[SCRRULES]	"SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/
[WAP-SI]	Open Mobile Alliance. Service Indication for WAP Push messaging. URL:http://www.openmobilealliance.org/tech/affiliates/wap/wap-167-serviceind-20010731-a.pdf
[WAP-SL]	Open Mobile Alliance. Service Loading for WAP Push Messaging. URL:http://www.openmobilealliance.org/tech/affiliates/wap/wap-168-serviceload-20010731-a.pdf
[W3C-FORMS]	"Use of Forms". URL:http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.2
[W3C-URLENC]	W3C HTML 2.0 Specification, form-urlencoded Media Type, URL: http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1
[XMLSchema1]	W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL: http://www.w3.org/TR/xmlschema-1/
[XMLSchema2]	W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL: http://www.w3.org/TR/xmlschema-2/

2.2 Informative References

[OMADICT]	"Dictionary for OMA Specifications", Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:http://www.openmobilealliance.org/
[REST_WP]	"White Paper on Guidelines for ParlayREST API specifications", Open Mobile Alliance™, OMA-WP-Guidelines-for-ParlayREST-API-specifications, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

[N/A]	[N/A]

3.3 Abbreviations

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
EMS	Enhanced Message Service
HTTP	HyperText Transfer Protocol
IM	Instant Message
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
MMS	Multi Media System
MSISDN	Mobile Subscriber ISDN Number
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
RTX	Ring Tone eXtended
SCR	Static Conformance Requirements
SI	Service Indication
SL	Service Logic
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 ParlayREST Technical Specification for Multimedia Messaging contains the HTTP protocol binding for the Parlay X Multimedia Messaging Web Services specification, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON, and form-urlencoding).

4.1 Version 1.0

Version 1.0 of the Multimedia Messaging Service ParlayREST API specification supports the following operations:

- Send message to a terminal
- Check delivery status of the outgoing message
- Check incoming messages (polling mode)
- Create subscriptions for notifications for inbound messages based on given criteria (online)
- Delete subscriptions for notifications for inbound messages (online)
- Create subscriptions for notification for outbound messages based on given criteria (online)
- Delete subscriptions for notification for outbound messages (online)
- Retrieve message content
- Confirm message retrieval by deleting message (execute delete command)

5. Messaging API definition

This section is organized to support a comprehensive understanding of the MMS API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

The terms “inbound” and “outbound” used in resource names and data structures refer to incoming, respectively outgoing messages from the client perspective. The term “subscription” refers to the online creation of resources (using requests in this specification). The term “registration” refers to the offline creation of resources using mechanisms out of scope of this specification. The resources created during registrations as well as subscriptions can generate notifications, for example about the delivery status of outgoing messages (subscription), or about incoming messages (registration).

Common data types, naming conventions, fault definitions and namespaces are defined in [REST_TS_Common].

The remainder of this document is structured as follows:

Section 5 starts with a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). In addition, for each supported resource/verb combination, the table lists the Parlay X equivalent operation, where applicable. What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

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

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

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

5.1 Resources Summary

This section summarizes all the resources used by the Messaging API. The resources are defined with the goal of supporting unified messaging, to allow their re-use by other APIs.

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.

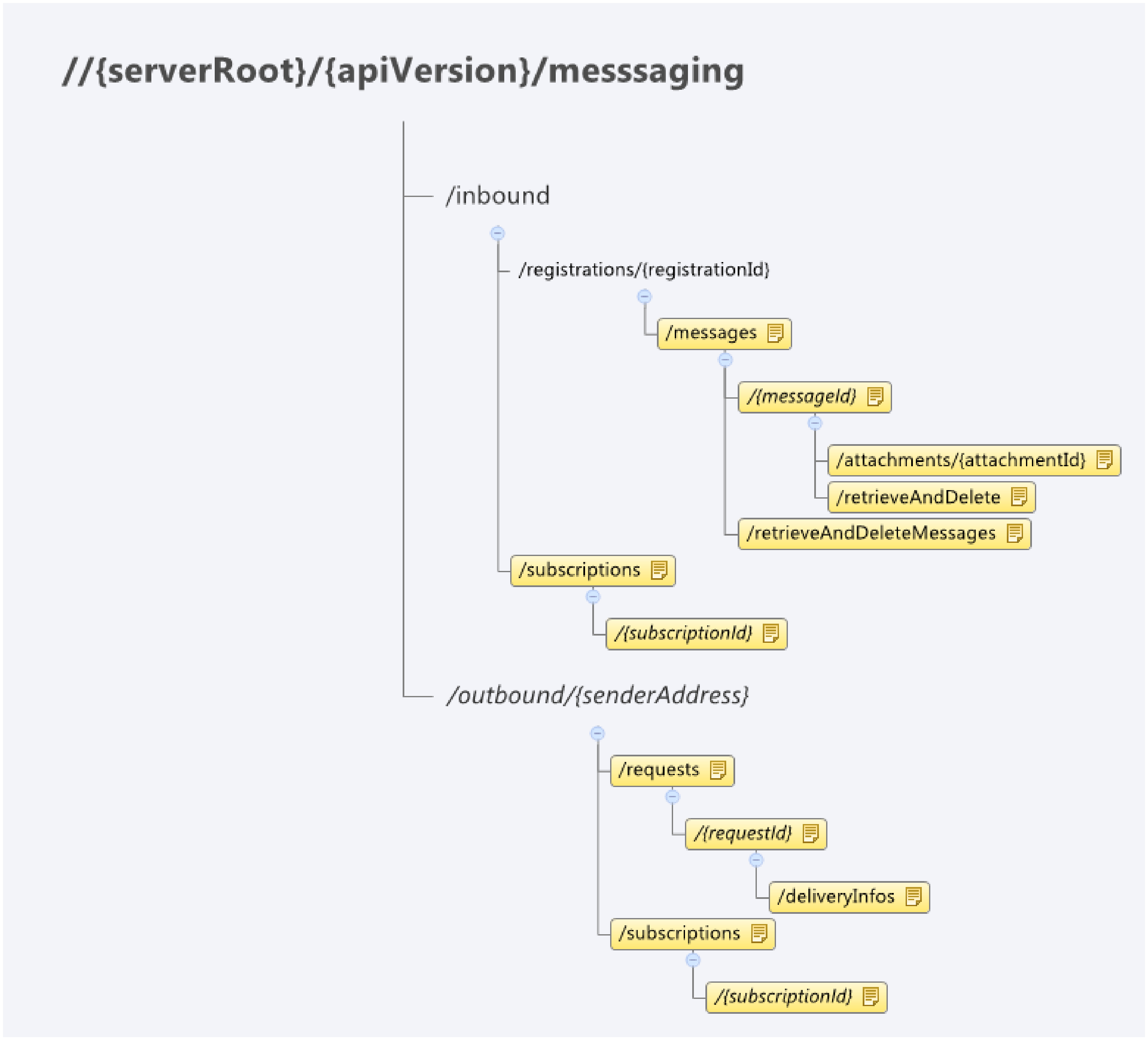
//{serverRoot}/{apiVersion}/messsaging
/inbound
/registrations/{registrationId}
/messages
/{messageId}
/attachments/{attachmentId}
/retrieveAndDelete
/retrieveAndDeleteMessages
/subscriptions
/{subscriptionId}
/outbound/{senderAddress}
/requests
/{requestId}
/deliveryInfos
/subscriptions
/{subscriptionId}

Figure 1Resource structure defined by this specification

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods. The "PX" row indicates the Parlay X SOAP equivalent operation.

Purpose: Inbound multimedia messages for periodic polling (based on a provisioning step configuration)

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Inbound messages for a given registration	/inbound/registrations/{registrationId}/messages Note: Used by clients that periodically poll for incoming messages. Retrieval criteria have to be provisioned in advance.	InboundMessageList	read one or more messages from gateway storage	no	no	no
			PX: GetReceivedMessages			
Inbound messages retrieve and delete using registration	/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages	InboundMessageList InboundMessageRetrieveAndDeleteRequest	no	pops one or more messages from the gateway storage (removes it if successful)	no	no
				PX: GetReceivedMessages		
Retrieval and deletion of individual inbound message using registration	/inbound/registrations/{registrationId}/messages/{messageId }/retrieveAndDelete	InboundMessage InboundMessageRetrievalAndDeleteRequest	no	pops one message and all attachments at once (mime formated) from the gateway storage (removes it if successful)	no	no
				PX: GetReceivedMessages		
Inbound message for a given	/inbound/registrations/{registrationId}/messages/{messageId }	InboundMessage	read one message from gateway	no	no	delete one message from

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
registration			storage			gateway storage
			PX: GetMessage GetMessageURIs			PX: GetReceivedMessages
Inbound message attachment	/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId}	Any MIME content (the one of the attachment)	read individual message attachment	no	no	delete attachment from gateway
			No PX equivalent			No PX equivalent

Purpose: Subscription Management for Inbound messages

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Inbound message subscriptions	/inbound/subscriptions	SubscriptionList	read all active subscriptions	create new message subscription	no	no
			No PX equivalent	PX: StartMessageNotification		
Individual inbound message subscription	/inbound/subscriptions/{subscriptionId}	Subscription	read individual subscription	no	no	removes subscription and stops corresponding message notifications
			No PX equivalent			PX: StopMessageNotification

Purpose: Callback notifications for Inbound Messages

Resource	URL <specified by the client>	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Client notification about inbound message	<specified by the client when subscription is created or during provisioning process>	InboundMessageNotification	no	notifies client about new inbound message	no	no
				PX: NotifyMessageReception		

Purpose: Sending message and obtaining the Delivery Status

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Outbound message requests	/outbound/{senderAddress}/requests	OutboundMessageRequestList OutboundMessageRequest	read all pending outbound message reference with current delivery status	create new outbound messages request	no	no
			No PX equivalent	PX: SendMessage		
Outbound message request and delivery status	/outbound/{senderAddress}/requests/{requestId}	OutboundMessageRequest	read message and delivery status for the individual outbound message request	no	no	no
			No PX equivalent			
Outbound message delivery status	/outbound/{senderAddress}/requests/{requestId}/deliveryInfos	DeliveryInfoList	read delivery status for the individual	no	no	no

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
			outbound message request			
			PX: GetMessageDeliveryStatus			

Purpose: Subscription Management for Outbound Message Delivery Status (overwrites individual request notifications)

Resource	URL Base URL: http://{serverRoot}/{apiVersion}/messaging	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Outbound message delivery notification subscriptions	/outbound/{senderAddress}/subscriptions	DeliveryReceiptSubscriptionList DeliveryReceiptSubscription	read all outbound message subscriptions	create new delivery receipt subscription	no	no
			No PX equivalent	PX: StartDeliveryReceiptNotification		
Individual outbound message delivery notification subscription	/outbound/{senderAddress}/subscriptions/{subscriptionId}	DeliveryReceiptSubscription	read an individual outbound message subscription	no	no	remove delivery recept notification subscription and stop corresponding delivery receipt notifications
			No PX equivalent			PX: StopDeliveryReciptNotification

Purpose: Callback notifications for Outbound Message Delivery Status

Resource	URL <specified by the client>	Data Structures	HTTP verbs			
			GET	POST	PUT	DELETE
Client notification about outbound message delivery status	<specified by the client when outbound request is submitted>	DeliveryInfoNotification	no	Notifies client about delivery status of outgoing requests	no	no
				PX: NotifyMessageDeliveryReceipt		

5.2 Messaging (MMS, SMS, WAP, IM) ParlayREST API Data Structures

The namespace for the Messaging data types is:

urn:oma:xml:rest:messaging:1

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace is used in the present document to refer to the data types defined in [REST_TS_Common]. The use of the names 'xsd' and 'common' is not semantically significant.

5.2.1 Type: InboundMessageList

Received message list.

Element	Type	Optional	Description
inboundMessage	InboundMessage [0..unbounded]	Yes	It may contain an array of messages received according to the specified registrationid.
totalNumberOfPendingMessages	xsd:int	Yes	Total number of messages in the gateway storage waiting for retrieval at the time of the request
numberOfMessagesInThisBatch	xsd:int	Yes	Number of the messages included in the response (part of the totalNumberOfPendingMessages)
resourceURL	xsd:anyURI	No	Self referring URL

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

5.2.2 Type: InboundMessage

Individual incoming message.

Element	Type	Optional	Description
destinationAddress	xsd:anyURI	No	Number associated with the invoked Message service, i.e. the destination address used by the terminal to send the message.
senderAddress	xsd:anyURI	No	Indicates message senderAddress.
dateTime	xsd:dateTime	Yes	Time when message was received by operator
resourceURL	xsd:anyURI	Yes	Self referring URL
link	common:Link[0..unbounded]	Yes	Link to other resources that are in relationship with the resource

messageId	xsd:string	Yes	OPTIONAL server generated message identifier. This field MUST be present when the type of the message differs from a plain text SMS, i.e. the element in the choice below has a type other than InboundSMSTextMessage.
inboundSMSTextMessage	InboundSMSTextMessage	Choice	Inbound SMS Text Message
inboundMMSMessage inboundIMMessage	InboundMMSMessage InboundIMMessage	Choice Choice	Inbound MMS Message Inbound IM Message

XSD modelling uses a "choice" to select either a inboundSMSTextMessage, inboundMMSMessage or inboundIMMessage.

5.2.3 Type: InboundMessageNotification

Notification carrying an individual incoming message.

Element	Type	Optional	Description
callbackData	xsd:string	Yes	CallbackData as passed by the application during the associated Send MMS operation. See [REST_TS_Common], section 6.2.4.
inboundMessage	InboundMessage	No	Multimedia message
link	common:Link[0..unbounded]	Yes	Link to other resources. For example we can have a link to the subscription used to receive this message.

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

5.2.4 Type: InboundSMSTextMessage

Element	Type	Optional	Description
message	xsd:string	No	Short message content.

5.2.5 Type: InboundMMSMessage

Element	Type	Optional	Description

subject	xsd:string	Yes	If present, indicates the subject of the received message.
priority	MessagePriority	Yes	The priority of the message: default is Normal.
link	common:Link[0..unbounded]	Yes	Link to other resources (like individual attachments: <Link rel="attachment" href="../inbound/registration/{registrationId} /messages/{messageId }/attachments/{attatchmentId}"/>)
bodyText	xsd:string	Yes	Contains the message body if it is encoded as ASCII text

5.2.6 Type: InboundIMMessage

Element name	Element type	Optional	Description
subject	xsd:string	Yes	If present, indicates the subject of the received IM message.
priority	MessagePriority	Yes	The priority of the message: default is Normal.
link	common:Link[0..unbounded]	Yes	Link to other resources (like individual attachments: <Link rel="attachment" href="../inbound/registration/{registrationId} /messages/{messageId }/attachments/{attatchmentId}"/>)
imFormat	IMFormat	Yes	If present, indicates the type of the received IM message. Otherwise any IM message type could be assumed (for example, server could not determine what type the received IM message is)
bodyText	xsd:string	Yes	Contains the message body if it is encoded as ASCII text

5.2.7 Type: SubscriptionList

Element	Type	Optional	Description
subscription	Subscription[0..unbounded]	Yes	It may contain an array of Subscription
resourceURL	xsd:anyURI	Yes	Self referring URL

A root element named subscriptionList of type SubscriptionList is allowed in request and/or response bodies

5.2.8 Type: Subscription

Element	Type	Optional	Description
callbackReference	common:CallbackReference	No	Client's Notification endpoint and parameters
destinationAddress	xsd:anyURI [1…unbounded]	No	the destination address of the multimedia message
criteria	xsd:string	Yes	The text to match against to determine the application to receive the notification. This text is matched against the first word, as defined as the initial characters after discarding any leading Whitespace and ending with a Whitespace or end of the string. The matching shall be case-insensitive. If the subject of the multimedia message is present it shall be used as the string, if not the string is defined as the first plain/text part of the content [3GPP TS 23.140]
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
resourceURL	xsd:anyURI	Yes	Self referring URL
link	common:Link[0..unbounded]	Yes	Link to other resources that are in relationship with the resource
useAttachmentURLs	xsd:boolean	Yes	Default: false If set to 'true', inbound message has links to attachments. Otherwise, inbound message includes attachments using MIME

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

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

5.2.9 Type: InboundMessageRetrieveAndDeleteRequest

Element	Type	Optional	Description
retrievalOrder	RetrievalOrder	Yes	Specifies order in which messages should be retrieved if there are more then one pending
priority	MessagePriority	Yes	The priority of the message: default is Normal.
maxBatchSize	xsd:int	Yes	Specifies maximum number of messages to be returned in the response
useAttachmentURLs	xsd:boolean	Yes	Default: false If set to ‘true’, inbound message will have links to attachments. Otherwise, inbound message includes attachments using MIME

A root element named inboundMessageRetrieveAndDeleteRequest of type InboundMessageRetrieveAndDeleteRequest t is allowed in request and/or response bodies

5.2.10 Type: OutboundMessageRequestList

Element	Type	Optional	Description
outboundMessageRequest	OutboundMessageRequest [0..unbounded]	Yes	Outbound message requests that have been sent by the application and still exist in the server. Message requests usually exist on the server for a little time after reaching their final Delivery Status
resourceURL	xsd:anyURI	Yes	Self referring URL

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

5.2.11 Type: OutboundMessageRequest

Element	Type	Optional	Description
address	xsd:anyURI [1..unbounded]	No	Destination addresses for the Message.
senderAddress	xsd:anyURI	No	User Identity of the Sender of the message. The associated MSISDN

			will appear in the receiver terminal, unlesss a senderName is specificated. This parameter shall match the User's Identity included in the Authorization header.
senderName	xsd:string	Yes	Name of the sender to appear on the user's terminal as the originator of the message. If this parameter is used, a set of allowed values shall be set during provisioning each sender (i.e.: for each User provisioned in the System).
charging	common:Charging Information	Yes	Charging to apply to this message.
receiptRequest	common:CallbackReference	Yes	It defines the notification endpoint and parameters that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible.
outboundSMSTextMessage	OutboundSMSTextMessage	Choice	Included if a SMSText is being sent.
outboundSMSLogoMessage	OutboundSMSLogoMessage	Choice	Included if a SMSLogo is being sent.
outboundSMSRingToneMessage	OutboundSMSRingToneMes sage	Choice	Included if a SMSRingtone is being sent.
outboundWAPMessage	OutboundWAPMessage	Choice	Included if WAP is being used
outboundMMSMessage	OutboundMMSMessage	Choice	Included if MMS is being sent.
ouboundIMMessage	OutboundIMMessage	Choice	Included if IM is being sent
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.

resourceURL	xsd:anyURI	Yes	Self referring URL
link	common:Link[0..unbounded]	Yes	Link to other resources that are in relationship with the resource
deliveryInfoList	DeliveryInfoList	Yes	The Delivery Information (filled in by the server)

XSD modelling uses a “choice” to select outboundSMSTextMessage, outboundSMSLogoMesage, outboundSMSRingToneMessage, outboundWAPMessage, outboundMMSMessage or outboundIMMessage.

A root element named outboundMessageRequest of type OutboundMessageRequest is allowed in request and/or response bodies

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

5.2.12 Type: OutboundMMSMessage

Element	Type	Optional	Description
subject	xsd:string	Yes	If present, indicates the subject of the received message.
priority	MessagePriority	Yes	The priority of the message: default is Normal.

5.2.13 Type: OutboundWAPMessage

Element	Type	Optional	Description
contentType	WAPContent	No	The type of content delivery notification to send.
targetURL	xsd:anyURI	No	A URL from which content may be loaded by a terminal
serviceLoadingAction	ServiceLoadingAction	Choice	There is no user intervention. If the parameter is not specified, the default value will be “ExecuteLow”. See [WAP-SL] for more details. May be present only if ContentType is “ServiceLoading”
serviceIndicationAction	ServiceIndicationAction	Choice	Allows controlling the level of intrusiveness, of outbound wap push messages. According to [WAP-SI] it contains a text string specifying the action to be taken when the message is received.

			If the parameter is not specified, the value "SignalMedium" is used May be resent only if ContentType is “ServiceIndicatiion”
text	xsd:string	Yes	Information that accompanies the push. May be present only if ServiceIndicationAction is present and ContentType is “ServiceIndicatiion”
created	xsd:dateTime	Yes	This attribute may be used to specify the date and time associated with the creation or last modification of the content indicated by targetURL, which may differ from the date and time when the message was created. May be present only if ContentType is “ServiceIndicatiion”.

XSD modelling uses a “choice” to select either a serviceLoadingAction or serviceIndicationAction plus text and created.

5.2.14 Type: OutboundSMSTextMessage

Element	Type	Optional	Description
message	xsd:string	No	Short message content.

5.2.15 Type: OutboundSMSLogoMessage

Element	Type	Optional	Description
image	xsd:base64Binary	No	The image in jpeg, gif or png format. The image will be scaled to the proper format
smsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'

5.2.16 Type: OutboundSMSRingToneMessage

Element	Type	Optional	Description
ringTone	xsd:string	No	The ring-tone in RTX format.

			Note:In the RTX Ringtone Specification,an RTX file is a text file, containing the ring-tone name, a control subclause and a subclause containing a comma separated sequence of ring tone commands.
smsFormat	SmsFormat	No	Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'

5.2.17 Type: OutboundIMMessage

Element	Type	Optional	Description
subject	xsd:string	Yes	If present, indicates the subject of the received message.
imFormat	IMFormat	Yes	The type of IM
bodyText	xsd:string	Yes	Contains the message body if it is encoded as ASCII text

5.2.18 Type: DeliveryInfoList

Element	Type	Optional	Description
resourceURL	xsd:anyURI	No	Self referring URL
link	common:Link[0..unbounded]	Yes	Linked to other resources that are in relationship with the resource
deliveryInfo	DeliveryInfo[1…unbounded]	No	Delivery Information

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

5.2.19 Type: DeliveryInfoNotification

Element	Type	Optional	Description
callbackData	xsd:string	Yes	CallbackData if passed by the application in the receiptRequest element during the associated Send SMS operation. See [REST_TS_Common], section 6.2.4.
deliveryInfo	DeliveryInfo [1..unbounded]	No	Delivery Information
link	common:Link[0..unbounded]	Yes	Link to other resources. For example

			we can have a link to the original outbound message request.

5.2.20 Type: DeliveryInfo

Element	Type	Optional	Description
address	xsd:anyURI	No	Outbound message destination address
deliveryStatus	DeliveryStatus	No	Indicates the delivery result for the destination address.
description	xsd:string	Yes	Used together with Delivery Status (e.g.DeliveryImpossible) to provide additional information.
link	common:Link[0..unbounded]	Yes	Link to other resources. For example we can have a link to the original outbound message request.

5.2.21 Type: DeliveryReceiptSubscriptionList

Element	Type	Optional	Description
resourceURL	xsd:anyURI	No	Self referring URL
link	common:Link[0..unbounded]	Yes	Link to other resources that are in relationship with the resource
deliveryReceiptSubscription	DeliveryReceiptSubscription[0…unbounded]	Yes	Delivery Information

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

5.2.22 Type: DeliveryReceiptSubscription

Element	Type	Optional	Description
callbackReference	common:CallbackReference	No	Notification endpoint and parameters definition
filterCriteria	xsd:string	No	The FilterCriteria will allow the service to filter flexibly. One example would be for the Service Provider to filter based on first 4 digits in MSISDN. This however is implementation specific and will be left to the Service Provider.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
resourceURL	xsd:anyURI	Yes	Self referring URL
link	common:Link[0..unbounded]	Yes	Link to other resources that are in relationship with the resource

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

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

5.2.23 Enumeration: DeliveryStatus

Enumeration	Description
DeliveredToTerminal	Successful delivery to Terminal.
DeliveryUncertain	Delivery status unknown: e.g. because it was handed off to another network.
DeliveryImpossible	Unsuccessful delivery; the message could not be delivered before it expired.
MessageWaiting	The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.
DeliveredToNetwork	Successful delivery to the network enabler responsible for distributing the multimedia message further in the network.
DeliveryNotificationNotSupported	Unable to provide delivery receipt notification. NotifyMessageDeliveryReceipt function will provide DeliveryNotificationNotSupported to indicate that delivery receipt for the

	specified address in a SendMessage is not supported.

5.2.24 Enumeration: IMFormat

Enumeration	Description
IM	Instant (immediate) messaging service (Can be short IM or large IM. Underlying network can decide message type from message context)
IMPagerMode	Short IM text message, as defined in [OMA-IM-TS].
IMLargeMode IMFileTransfer	Large IM message with multimedia, as defined in [OMA-IM-TS]. Large IM used for File Transfer, as defined in [OMA-IM-TS]

5.2.25 Enumeration: MessagePriority

Enumeration	Description
Default	Default message priority
Low	Low message priority
Normal	Normal message priority
High	High message priority

5.2.26 Enumeration: RetrievalOrder

Enumeration	Description
OldestFirst	Retrieve in the order from oldest to newest
NewestFirst	Retrieve in the order from newest to oldest

5.2.27 Enumeration: ServiceIndicationAction

Enumeration	Description
SignalNone	The message MUST NOT be presented or postponed. If anything, only the info part could be used by the client for some purpose [WAP-SI].
SignalLow	The SI MUST be postponed without user intervention
SignalMedium	The SI MUST be presented as soon as the implementation allows that to be carried out in a non-user-intrusive manner.
SignalHigh	The SI MUST be presented as soon as the implementation allows that to be carried out in a non-user-intrusive manner, or earlier if considered appropriate (which MAY result in a user-intrusive behaviour). This decision can either be based on user preference settings or be carried out at the discretion of the implementation.
Delete	The message should be discarded.

5.2.28 Enumeration: ServiceLoadingAction

Enumeration	Description
ExecuteLow	The service identified by the URI provided by the SL's href attribute is loaded in the same way as the user agent otherwise performs method requests initiated by the end-user. This implies that service content is fetched either from an origin server or from the client's cache, if available. Once the method request is successfully completed, the user agent loads the service into a clean user agent context and executes it. This MUST be carried out in an non-user-intrusive manner[WAP-SL]
ExecuteHigh	The service is loaded and executed in the same way as for ExecuteLow, but MAY result in a user-intrusive bahavior.
Cache	The service is loaded in the same way as for ExecuteLow. However, instead of executing the service (as described above) it is placed in the cache of the client. If no cache exists, the SL MUST be silently discarded.

5.2.29 Enumeration: SmsFormat

Enumeration	Description
Ems	EMS conversion
SmartMessaging	SmartMessaging conversion

5.2.30 Enumeration: WAPContent

Enumeration	Description
ServiceIndication	The Service Indication (SI) content type provides the ability to send notifications to end-users in an asynchronous manner. In its most basic form, an SI contains a short message and a URI indicating a service. The message is presented to the end-user upon reception, and the user is given the choice to either start the service indicated by the URI immediately, or postpone the SI for later handling. [WAP-SI]
ServiceLoading	The Service Loading (SL) content type provides the ability to cause a user agent on a mobile client to load and execute a service. The SL contains a URI indicating the service to be loaded by the user agent without user intervention when appropriate. [WAP-SL]

5.2.31 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):

- InboundMessage
- InboundMessageList
- Subscription

- SubscriptionList
- OutboundMessageRequest
- OutboundMessageRequestList
- DeliveryInfoList
- DeliveryReceiptSubscription
- DeliveryReceiptSubscriptionList
- attachment

These values indicate the kind of resource that the link points to. The value “attachment” indicates that the Link refers to an attachment of the message.

5.3 Sequence Diagrams

5.3.1 Send message and check the delivery status

This figure below shows a scenario for sending a multimedia message and get the delivery status of the message.

The resources:

- To send a multimedia message, create new resource under

  http://{serverRoot}/{apiVersion}/messaging/{senderAddress}/outbound/requests

- To get the delivery status of the message, do either a. or b.

  a. read the newly created resource including the delivery status of the message

     http://{serverRoot}/{apiVersion}/messaging/{senderAddress}/outbound/requests/{requestId}

  b. directly read the resource

     http://{serverRoot}/{apiVersion}/messaging/{senderAddress}/outbound/requests/{requestId} /deliveryInfos

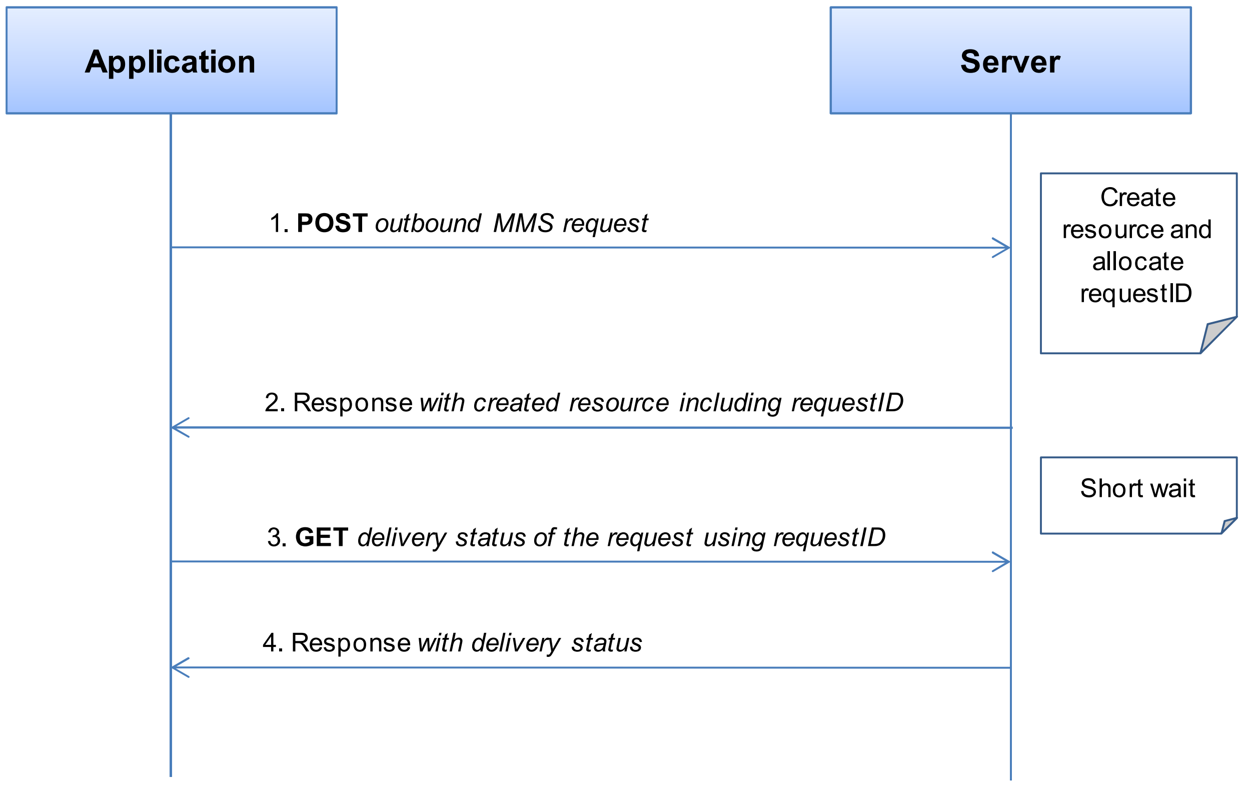
Application
Server
1. POST outbound MMS request
Create resource and allocate requestID
2. Response with created resource including requestID
Short wait
3. GET delivery status of the request using requestID
4. Response with delivery status

Figure 2 Send message and check the delivery status

Outline of the flows:

1. An application initiates the creation of new outbound message request using POST and receives the created request resource with a resource URL containing the requestId.
2. The application requests the resource of the sent message with the given resource URL (containing the requestId) using GET and optionally gets the delivery status, or
3. The application requests the delivery status of the sent message with the given delivery info list URL using GET and gets the status.

5.3.2 Inbound message delivery (push mode)

This figure below shows a scenario for starting notification of inbound message with specific criteria on-line and receiving it when the message having the specified criteria arrives.

The resources:

- To start subscription to notifications for inbound messages, create new resource under

  http://{serverRoot} /{apiVersion}/messaging/inbound/subscriptions
- To notify the application about the message arrival, POST notification to the client supplied notifyURL
- To stop the subscription to notifications, delete the resource

  http://{serverRoot} /{apiVersion}/messaging/inbound/subscriptions/{subscriptionId}

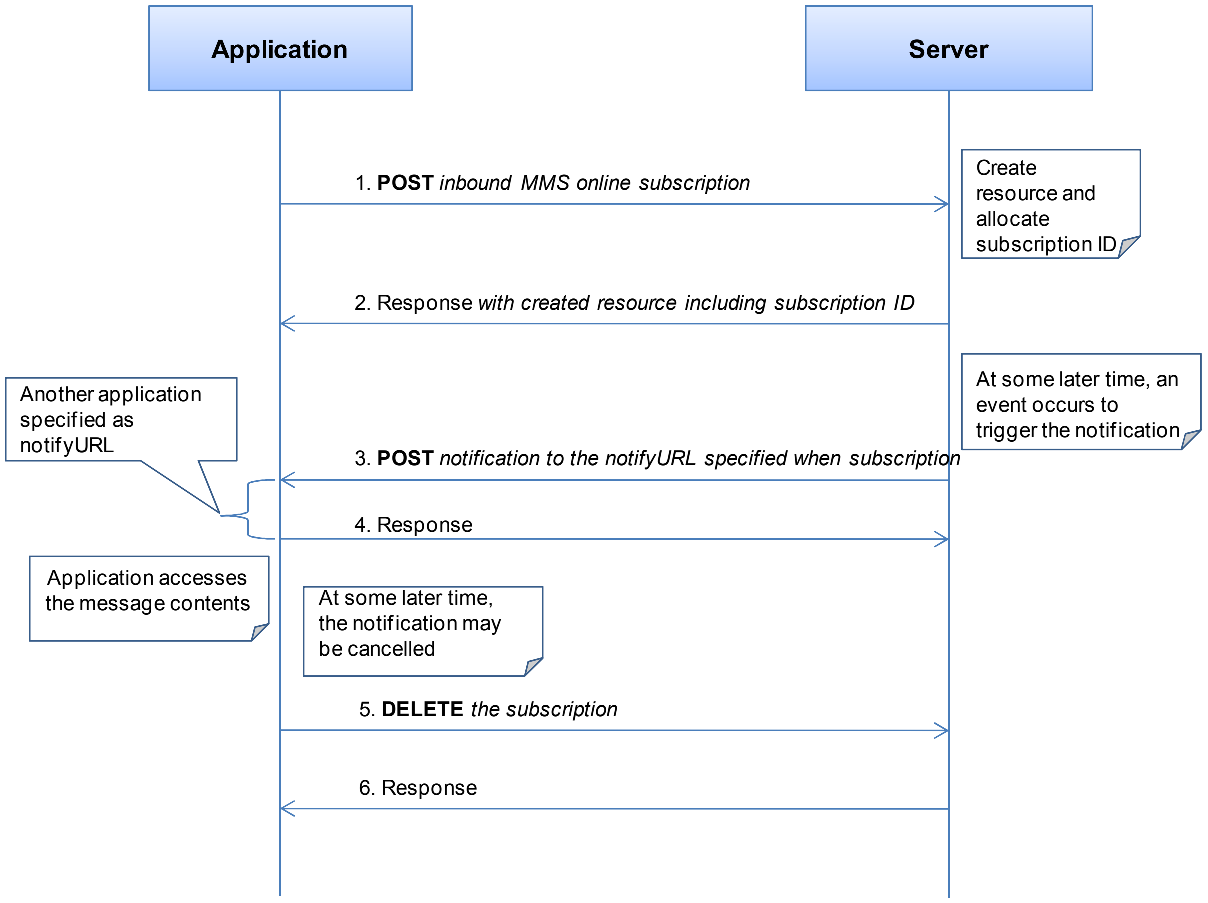
Application
Server
1. POST inbound MMS online subscription
Create resource and allocate subscription ID
2. Response with created resource including subscription ID
At some later time, an event occurs to trigger the notification
Another application specified as notifyURL
3. POST notification to the notifyURL specified when subscription
4. Response
Application accesses the message contents
At some later time, the notification may be cancelled
5. DELETE the subscription
6. Response

Figure 3 Inbound message delivery (push mode)

Outline of the flows:

1. An application subscribes to notifications for inbound messages using POST and receives the resulting resource URL containing the subscriptionId.
2. When the message which satisfies the specified criteria arrives, the REST service on the server notifies the application of the message arrival using POST so that the application may read the message request.
3. The application reads the attached content using attachmentURL in the message request.
4. The application stops the notifications subscription using DELETE with a resource URL containing the subscriptionId.

5.3.3 Inbound message delivery (polling mode)

This figure below shows a scenario for checking for incoming messages using retrieval criteria that is set up offline ,getting one message ,and deleting it from the storage.

The resources:

- To retrieve incoming messages satisfying the criteria set up in advance, get the resource

  http://{serverRoot} /{apiVersion}/messaging/inbound/registrations/{registrationId}/messages

  This will return message references (identifiers and if requested, attachments URLs).

- To read one message from the storage, get the resource

  http://{serverRoot} /{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}

  This will return the whole message (MIME format)

- To read individual attachments of an message, based on message identifiers and attachment URLs:

  http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}/attachments/{attachmentId}

- To remove one message from the storage, delete the resource

  http://{serverRoot} /{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}

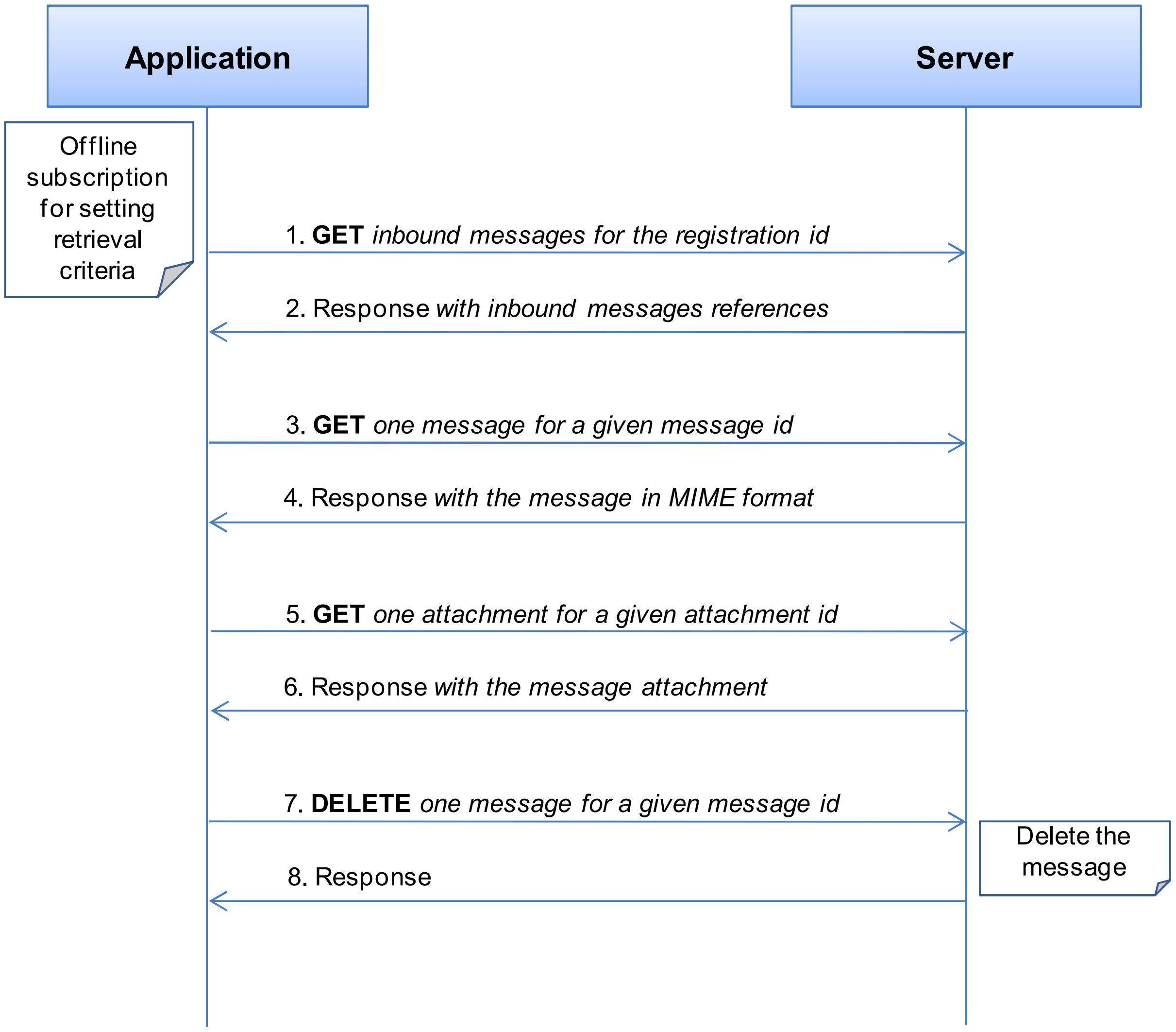
Application
Server
Offline subscription for setting retrieval criteria
1. GET inbound messages for the registration id
2. Response with inbound messages references
3. GET one message for a given message id
4. Response with the message in MIME format
5. GET one attachment for a given attachment id
6. Response with the message attachment
7. DELETE one message for a given message id
Delete the message
8. Response

Figure 4 Inbound message delivery (polling mode)

Outline of the flows:

1. In advance, the notification of message reception with specific criteria is registered offline.
2. An application requests the list of the incoming messages fulfilling specified criteria using GET with a resource URL containing the registrationId.

3. The application reads one message request using GET.with a resource URL containing the messageId
4. The application reads one attachment to the message using GET with a resource URL containing the attachmentId.
5. The application removes one of the messages from gateway storage using DELETE with a resource URL containing the messageId .

5.4 Resource: Inbound messages for a given registration

The resource used is:
http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages

This resource is for polling incoming messages using retrieval criteria that is setup in advance during provisioning process for a particular application.

5.4.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API clients wants to use
registrationId	reference to the retrieval criteria provisioned in advance and known to the client application. Analog of ParlayX registrationIdentifier

5.4.2 Response Codes

5.4.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.4.2.2 Exception fault codes

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

5.4.3 GET

This operation is used for reliable inbound message retrieval for the particular client. Messages will remain on the server until client will confirm successful retrieval by executing DELETE command (see DELETE on Inbound message.

Note: ParlayX SOAP equivalent is GetReceivedMessages, but not quite the same because DELETE is required for confirmation (see DELETE on Inbound message).

Request URL parameters are:

Name	Type/Values	Optional	Description
maxBatchSize	xsd:int	Yes	Specifies maximum number of messages to be returned in the response
retrievalOrder	RetrievalOrder	Yes	Specifies order in which messages should be retrieved is

			there are more then one pending
useAttachmentURLs	xsd:boolean	Yes	Default: false If set to 'true', inbound message would have links to attachments. Otherwise, only message identifier will be returned, so that individual message retrieval can be done.
priority	MessagePriority	Yes	The priority of the messages to poll from the gateway. All messages of the specified priority and higher will be retrieved. If not specified, all messages shall be returned, i.e. the same as specifying Low.

5.4.3.1 Examples 1: useAttachmentURLs=false (Informative)

5.4.3.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2 HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.4.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}</resourceURL>
    <messageId>{messageId1}</messageId>
    <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN3</destinationAddress>
    <senderAddress>MSISDN4</senderAddress>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId2}</resourceURL>
    <messageId>{messageId2}</messageId>
       <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <totalNumberOfPendingMessages>20</totalNumberOfPendingMessages>
<numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
 <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/
messages?maxBatchSize=2</resourceURL>
</mms:inboundMessageList>
```

5.4.3.2 Example 2: request with invalid id (Informative)

5.4.3.2.1 Request

```
GET .../{apiVersion}/messaging/inbound/registrations/registration123/messages?maxBatchSize=2 HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.4.3.2.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="self"   href="http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/registration123/messages?maxBatchSize=2" />
  <serviceException>
    <messageId>SVC0002</messageId>
    <text>Invalid input value. The requested registration id: %1 is not valid </text>
    <variables>registration123</variables>
  </serviceException>
</common:requestError>
```

5.4.3.3 Example 3: useAttachmentURLs=true (Informative)

5.4.3.3.1 Request

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2&useAttachmentURLs=true HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.4.3.3.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}</resourceURL>
    <messageId>{messageId1}</messageId>
     <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
      <link rel="attachment"
        href="http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{mesageId1}
        /attachments/{attachmentId1}" />
      <link rel="attachment"
       href="http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
        /attachments/{attachmentId2}" />
      <bodyText>See attached picture</bodyText>
    </inboundMMSMessage>
  </inboundMessage>
  <totalNumberOfPendingMessages>20</totalNumberOfPendingMessages>
  <numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/
{messageId1}/attachments</resourceURL>
</mms:inboundMessageList>
```

5.4.4 PUT

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

5.4.5 POST

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

5.4.6 DELETE

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

5.5 Resource: Inbound messages retrieve and delete using registration

The resource used is:
http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages

This resource is used for retrieving and deleting the list of incoming messages using retrieval criteria that is setup in advance (offline - during provisioning process: short codes, etc) for a particular client.

After this step, attachments or individual messages are still available for the individual retrieval.

5.5.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API clients wants to use

registrationId	reference to the retrieval criteria provisioned in advance and known to the client application. Analog of ParlayX registrationIdentifier

5.5.2 Response Codes

5.5.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.5.2.2 Exception fault codes

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

5.5.3 GET

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

5.5.4 PUT

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

5.5.5 POST

This operation retrieves one or more messages from the gateway storage for the particular client. If retrieval is successful, it will delete message from gateway.

Notes: POST is used because resource state would be altered as result of the execution. GET is not a good fit here because it has to be idempotent. Client guidelines:

1) should NOT be used for reliable message delivery (see GET for reliable delivery). This is an optional alternative to the use of GET and DELETE on the …/inbound/subscriptions resource.

2) Default number of messages that would be returned in one batch is controlled by server configuration.

3) Messages would be deleted from gateway storage following a successful POST, after a maximum time interval as defined by a service policy. Client needs to retrieve body of the message with all attachments by executing separate POST using URL provided in mime-url attribute.

Note: ParlayX SOAP equivalent is GetReceivedMessages.

Parameters are passed in the request body using the InboundMessageRetrieveAndDeleteRequest data structure.

5.5.5.1 Example: useAttachmentURLs=false (Informative)

5.5.5.1.1 Request

```
POST .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages  HTTP/1.1
Accept: application/xml
Content-Length: nnn
Content-Type: application/xml; charset=UTF-8
Host: example.com:80
```

```
<?xml version="1.0" encoding="UTF-8"?>
<InboundMessageRetrieveAndDeleteRequest>
   <retrievalOrder>OldestFirst</retrievalOrder>
</InboundMessageRetrieveAndDeleteRequest>
```

5.5.5.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
    <!--  resourceURL is not included because message is deleted from the server already  -->
    <messageId>{messageId1}</messageId>
    <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN3</destinationAddress>
    <senderAddress>MSISDN4</senderAddress>
    <!--  resourceURL is not included because message is deleted from the server already -->
    <messageId>{messageId2}</messageId>
    <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <totalNumberOfPendingMessages>20</totalNumberOfPendingMessages>
<numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/
retrieveAndDeleteMessages</resourceURL>
</mms:inboundMessageList>
```

5.5.6 DELETE

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

5.6 Resource: Retrieval and deletion of individual inbound message using registration

The resource used is:
http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}/retrieveAndDelete

This resource is used to retrieve and simultaneously delete individual inbound message and all attachments stored by the gateway, in MIME representation. It is an alternative way to get access to the message. GET followed by delete on **http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId}** resource should be used for reliable delivery.

5.6.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API clients want to use
registrationId	reference to the retrieval criteria provisioned in advance and known to the client application. Analog of ParlayX registrationIdentifier
messageId	unique message identifier generated by server

5.6.2 Response Codes

5.6.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.6.2.2 Exception fault codes

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

5.6.3 GET

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

5.6.4 POST

This operation is used to read and delete one message from gateway storage. If successful, message would be deleted together with all associated attachments, after an agreed time interval as defined by a service policy.

Note: Equivalent ParlayX SOAP API is GetMessage.

5.6.4.1 Example (informative)

5.6.4.1.1 Request

```
POST .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/retrieveAndDelete HTTP/1.1
```

```
Accept: application/xml
Host: example.com:80
```

5.6.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Length: nnnnnn
Content-Type: multipart/form-data;
                boundary="==============123456==";
                type="application/xml"
MIME-Version: 1.0
Date: Thu, 04 Jun 2009 02:51:59 GMT

--==============123456==
Content-Disposition: form-data; name=”root-fields”
Content-Type: application/xml


<?xml version="1.0"?>
<mms:inboundMessage xmlns:mms="urn:oma:xml:rest:messaging:1">
  <destinationAddress>MSISDN1</destinationAddress>
<senderAddress>MSISDN2</senderAddress>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
   </resourceURL>
  <messageId>{messageId1}</messageId>
  <inboundMMSMessage>
    <subject>Who is RESTing on the beach?</subject>
  </inboundMMSMessage>
</mms:inboundMessage>

--==============123456==
Content-Disposition: form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”====aaabbb”
====aaabbb
Content-Disposition:attachments;filename=”textBody.txt”;
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8 bit


Look at the attached picture


====aaabbb
Content-Disposition:attachments;filename=”image1.gif”;
Content-Type: image/gif
MIME-Version: 1.0
Content-ID: <99334422@example.com>

GIF89a...binary image data...
--==============123456==--
```

5.6.5 PUT

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

5.6.6 DELETE

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

5.7 Resource: Inbound message for a given registration

The resource used is:
http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }

This resource provides access to individual inbound message stored by the gateway. Combination of GET/DELETE is used by clients that are polling incoming messages and require reliable delivery. Each message would have to be deleted separately as a confirmation of successful retrieval.

5.7.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API clients want to use
registrationId	reference to the provisioned in advance and known to the client application. Analogous to Parlay X registrationIdentifier
messageId	unique message identifier generated by server

5.7.2 Response Codes

5.7.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.7.2.2 Exception fault codes

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

5.7.3 GET

This operation is used to read one message from gateway storage. Message is not deleted. Delete command need to be executed to confirm delivery and free resources occupied by the message and associated attachments.

Note: Equivalent ParlayX SOAP API is GetMessage for MMS. In case the parameter "useAttachmentURLs" is set to "true", equivalent ParlayX SOAP API is GetMessageURIs.

5.7.3.1 Example: MMS (Informative)

5.7.3.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }?resFormat=XML HTTP/1.1
Host: example.com:80
```

5.7.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: : multipart/form-data; boundary=”=====12345====”
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT


====12345====
Content-Disposition=multipart/form-data; name=”root-fields”
Content-Type=application/xml
Content-Length: nnnn
<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessage xmlns:mms="urn:oma:xml:rest:messaging:1">
  <destinationAddress>MSISDN1</destinationAddress>
<senderAddress>MSISDN2</senderAddress>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
   </resourceURL>
  <messageId>{messageId1}</messageId>
  <inboundMMSMessage>
    <subject>Who is RESTing on the beach?</subject>
  </inboundMMSMessage>
</mms:inboundMessage>

====12345====
Content-Disposition: form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”====aaabbb”
====aaabbb
Content-Disposition:attachments;filename=”textBody.txt”;
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8 bit


Look at the attached picture


====aaabbb
Content-Disposition:attachments;filename=”image1.gif”;
Content-Type: image/gif
MIME-Version: 1.0
Content-ID: <99334422@example.com>

GIF89a...binary image data...
```

```
===============123456==
```

5.7.4 PUT

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

5.7.5 POST

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

Note: See Inbound MMS message retrieve and delete.

5.7.6 DELETE

Confirms message delivery and removes the message from the cache/storage on the gateway**.**

5.7.6.1 Example (Informative)

5.7.6.1.1 Request

```
DELETE .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId } HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.7.6.1.2 Response

```
HTTP/1.1 204 No content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.8 Resource: Inbound message attachment

The resource used is:

http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId}

This resource is used to provide access to individual MMS attachment stored by the gateway. Combination of GET/DELETE is used by clients that are polling incoming messages and require reliable delivery. Each attachment would have to be deleted separately as a confirmation of successful retrieval.

Individual deletions over all attachments would have the same effect as a DELETE over an individual message

(/inbound/registrations/{registrationId}/messages/{messageId}).

POST on ../retrieveAndDelete resource is used to pop (read and delete in the single step) MMS message (body+attachments) from the gateway storage. It would require no subsequent DELETE operations.

5.8.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)
registrationId	reference to the retrieval criteria provisioned in advance and known to the client application. Analog of ParlayX registrationIdentifier
messageId	unique message identifier generated by server
attatchmentId	unique attatchment identifier generated by server

5.8.2 Response Codes

5.8.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.8.3 GET

This operation is used to Read one MMS attachment from the gateway storage. Attachment is not deleted. Delete command need to be executed to confirm delivery and free resources occupied by the attachment.

Note: no equivalent ParlayX SOAP API for MMS attachment.

5.8.3.1 Example (Informative)

5.8.3.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId} HTTP/1.1
Accept: image/gif, image/png, image/jpeg, text/html, application/xml
Host: example.com:80
```

5.8.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Length: nnnnnn
Content-Type: image/gif
Date: Thu, 04 Jun 2009 02:51:59 GMT

...GIF89a...binary image data
```

5.8.4 PUT

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

5.8.5 POST

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

5.8.6 DELETE

This operation is used to confirm successful attachment retrieval and to remove it from the gateway storage.

5.8.6.1 Example (Informative)

5.8.6.1.1 Request

```
DELETE .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.8.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.9 Resource: Inbound message subscriptions

The resource used is: **http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions**

This resource gives access to inbound subscriptions for a particular client.

5.9.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API clients want to use (e.g. version 1 for 1.x)

5.9.2 Response Codes

5.9.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.9.2.2 Exception fault codes

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

5.9.3 GET

This operation is used to read active subscriptions for the particular client.

5.9.3.1 Example (Informative)

5.9.3.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.9.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:subscriptionList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <subscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/12345</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <destinationAddress>680180999</destinationAddress>
    <criteria>Urgent*</criteria>
    <clientCorrelator>567891</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/0000001</resourceURL>
    <useAttachmentURLs>false</useAttachmentURLs>
  </subscription>
  <subscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/54321</notifyURL>
      <callbackData>54321</callbackData>
     <notificationFormat>XML</notificationFormat>
    </callbackReference>
    <destinationAddress>80999</destinationAddress>
    <criteria>Urgent*</criteria>
    <clientCorrelator>567892</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/0000002</resourceURL>
    <useAttachmentURLs>false</useAttachmentURLs>
  </subscription>
</mms:subscriptionList>
```

5.9.4 PUT

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

5.9.5 POST

This operation is used to create a new inbound message subscription for the particular client.

Note: ParlayX SOAP equivalent is StartMessageNotification.

5.9.5.1 Example 1: returning a representation of created resource (Informative)

5.9.5.1.1 Request

```
POST .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<mms:subscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/88888</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>XML</notificationFormat>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
  <clientCorrelator>567893</clientCorrelator>
  <useAttachmentURLs>false</useAttachmentURLs>
</mms:subscription>
```

5.9.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<mms:subscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification//88888</notifyURL>
    <notificationFormat>XML</notificationFormat>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
   <clientCorrelator>567893</clientCorrelator>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions{subscriptionId1}</resourceURL>
  <useAttachmentURLs>false</useAttachmentURLs>
</mms:subscription>
```

5.9.5.2 Example 2: returning the location of created resource (Informative)

5.9.5.2.1 Request

```
POST .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<mms:subscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/88888</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>XML</notificationFormat>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
  <useAttachmentURLs>false</useAttachmentURLs>
</mms:subscription>
```

5.9.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:common:1">
  <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}</resourceURL>
</common:resourceReference>
```

5.9.6 DELETE

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

5.10 Resource: Individual inbound message subscription

The resource used is: **http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId}**

This resource controls individual subscription for inbound messages for a particular client.

5.10.1 Request URI variables

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

Name	Description

serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)
subscriptionId	identifies the subscription

5.10.2 Response Codes

5.10.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.10.2.2 Exception fault codes

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

5.10.3 GET

This operation is used to read an individual subscription for the particular client.

5.10.3.1 Example (Informative)

5.10.3.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.10.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:subscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/88888</notifyURL>
    <notificationFormat>XML</notificationFormat>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
<clientCorrelator>567893</clientCorrelator>
 <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId}</resourceURL>
  <useAttachmentURLs>false</useAttachmentURLs>
</mms:subscription>
```

5.10.4 PUT

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

5.10.5 POST

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

5.10.6 DELETE

This operation is used to delete a subscription for the particular client.

Note: Parlay X SOAP equivalent is StopMessageNotifications.

5.10.6.1 Example (Informative)

5.10.6.1.1 Request

```
DELETE .../{apiVersion}/messaging/inbound/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.10.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.11 Resource: Client notification about inbound message

This resource is a client provided callback URL for posting notifications about incoming messages. ParlayREST does not make any assumption about the structure of this URL.

5.11.1 Request URI variables

Client provided.

5.11.2 Response Codes

5.11.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.11.3 GET

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

5.11.4 PUT

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

5.11.5 POST

This operation is used to notify client about message arrival.

Note: ParlayX SOAP equivalent is NotifyMessageReception.

5.11.5.1 Example 1: Incoming MMS, include useAttachmentURLs=false (Informative)

5.11.5.1.1 Request

```
POST …/notifications/DeliveryInfoNotification/88888 HTTP/1.1
Accept: application/xml
Content-Length: 12345
Host: application.example.com:80
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageNotification xmlns:mms="urn:oma:xml:rest:messaging:1">
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
   <senderAddress>MSISDN2</senderAddress>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
   </resourceURL>
    <link rel=”Subscription”   href=”http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}”
    <messageId>{messageId1}</messageId>
    <inboundMMSMessage>
       <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
</mms:inboundMessageNotification>
```

5.11.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.11.5.2 Example 2: Incoming MMS, include useAttachmentURLs=true (Informative)

5.11.5.2.1 Request

```
POST …/notifications/DeliveryInfoNotification HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: 12345
Host: application.example.com:80
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageNotification xmlns:mms="urn:oma:xml:rest:messaging:1">
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
    </resourceURL>
    <link rel="Subscription" href=" http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}"
    <messageId>{messageId1}</messageId>
    <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
      <link rel="attachment"
          href="http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
           /attachments/{attachmentId1}" />
      <link rel="attachment"
        href="http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}/
        attachments/{attachmentId2}" />
      <bodyText>Look at the attached picture</bodyText>
   </inboundMMSMessage>
  </inboundMessage>
</mms:inboundMessageNotification>
```

5.11.5.2.2 Response

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

5.11.6 DELETE

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

5.12 Resource: Outbound message requests

The resource used is: **http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests**

This resource is used for sending outbound messages.

5.12.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)

senderAddress	sender application address. Typically SHORT CODE [REST_TS_COMMON], but could be a terminal address

5.12.2 Response Codes

5.12.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.12.2.2 Exception fault codes

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

5.12.3 GET

This operation is used to retrieve the list of pending outgoing requests.

5.12.3.1 Example (Informative)

5.12.3.1.1 Request

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.12.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:outboundMessageRequestList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <outboundMessageRequest>
    <address>tel:1350000001</address>
    <senderAddress>tel:1350000009</senderAddress>
      <outboundMMSMessage>
        <subject>Holiday greetings</subject>
      </outboundMMSMessage>
      <clientCorrelator>567894</clientCorrelator>
      <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId1}</resourceURL>
      <deliveryInfoList>
        <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId1}/deliveryInfos
        </resourceURL>
      <deliveryInfo>
        <address>tel:1350000001</address>
        <deliveryStatus>DeliveredToTerminal</deliveryStatus>
      </deliveryInfo>
    </deliveryInfoList>
  </outboundMessageRequest>
</mms:outboundMessageRequestList>
```

5.12.4 PUT

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

5.12.5 POST

This operation is used to create outgoing message request. It must follow the serialization guidelines described in section 5.6 of [REST_WP] in order to combine the multiple MIME body parts into the HTTP request message.

Note: ParlayX SOAP equivalent is SendMessage.

5.12.5.1 Example 1: regular request returning the representation of created resource (Informative)

5.12.5.1.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/form-data; boundary="===============123456==";
Host: example.com:80
MIME-Version: 1.0

--===============123456==
Content-Disposition: multipart/form-data; name="root-fields"
Content-Type: application/xml; charset="utf-8"
Content-Length: nnnn

<?xml version="1.0"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:1350000001</address>
  <address>tel:1350000999</address>
  <senderAddress>tel:1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </receiptRequest>
  <outboundMMSMessage>
    <subject>hello from the rest of us!</subject>
    <priority>High</priority>
  </outboundMMSMessage>
  <clientCorrelator>567895</clientCorrelator>
</mms:outboundMessageRequest>

--===============123456==

Content-Disposition: multipart/form-data; name="attachments"
Content-Type: multipart/mixed; boundary="===12345==="

===12345==="
Content-Disposition: attachments; filename="picture.gif"
Content-Type: text/plain;

See attached photo
```

```
===12345===”
Content-Disposition: attachment; filename="picture.gif"
Content-Type: image/gif

GIF89a...binary image data...
--===============123456==--
```

5.12.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:1350000001</address>
  <address>tel:1350000999</address>
  <senderAddress>tel:1351111999</senderAddress>
  <senderName>MyName</senderName>
   <outboundMMSMessage>
      <subject>Holiday greeings</subject>
   </outboundMMSMessage>
   <clientCorrelator>567895</clientCorrelator>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
  <requestId>{requestId}</requestId>
</mms:outboundMessageRequest>
```

5.12.5.2 Example 2: regular request returning the location of created resource (Informative)

5.12.5.2.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/form-data; boundary="===============123456==";
Host: example.com:80
MIME-Version: 1.0

--===============123456==
Content-Disposition: multipart/form-data; name=”root-fields”
Content-Type: application/xml; charset="utf-8"
Content-Length: nnnn

<?xml version="1.0"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:1350000001</address>
  <address>tel:1350000999</address>
  <senderAddress>tel:1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </receiptRequest>
  <outboundMMSMessage>
    <subject>hello from the rest of us!</subject>
    <priority>High</priority>
  </outboundMMSMessage>
  <clientCorrelator>567895</clientCorrelator>
</mms:outboundMessageRequest>

--===============123456==

Content-Disposition: multipart/form-data; name="attachments"
Content-Type: multipart/mixed; boundary="===12345==="

===12345==="
Content-Disposition: attachments; filename="picture.gif"
Content-Type: text/plain;

See attached photo

===12345==="
Content-Disposition: attachment; filename="picture.gif"
Content-Type: image/gif

GIF89a...binary image data...
--===============123456==--
```

5.12.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:common:1">
  <resourceURL> http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
</common:resourceReference>
```

5.12.5.3 Example 3: request with charging (Informative)

5.12.5.3.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/related; boundary="===============123456==";type="application/xml"
Host: example.com:80
MIME-Version: 1.0
```

```
--===============123456==
Content-Type: application/xml; charset="utf-8"

<?xml version="1.0"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:1350000001</address>
  <address>tel:1350000999</address>
  <senderAddress> tel:1351111999</senderAddress>
  <senderName>MyName</senderName>
  <charging>
    <description>Sample text for the charging information</description>
  </charging>
  <receiptRequest><!-- this is optional -->
    <notifyURL>http://example-application.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </receiptRequest>
  <outboundMMSMessage>
    <subject>hello from the rest of us!</subject>
    <priority>High</priority>
  </outboundMMSMessage>
  <clientCorrelator>567896</clientCorrelator>
</mms:outboundMessageRequest>

--===============123456==

Content-Disposition: multipart/form-data; name=”attachments”
Content-Type: multipart/mixed; boundary=”===12345===”

===12345===”
Content-Disposition: attachments; filename="picture.gif"
Content-Type: text/plain;
Content-Length: nnnn

See attached photo

===12345===”
Content-Disposition: attachment; filename="picture.gif"
Content-Type: image/gif
Content-Length: nnnn

GIF89a...binary image data...
--===============123456==--
```

5.12.5.3.2 Response for charging not supported

```
HTTP/1.1 400 Bad request
Content-Type: application/xml
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

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

```
  <policyException>
    <messageId>POL0008</messageId>
    <text>Charging is not supported</text>
  </policyException>
</common:requestError>
```

5.12.6 DELETE

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

5.13 Resource: Outbound message request and delivery status

The resource used is: **http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}**

This resource is used to retrieve an outbound message request including the message delivery status.

5.13.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)
senderAddress	identifies client application. Typically SHORT CODE [REST_TS_COMMON], but could be a terminal address
requestId	outbound message request identifier generated by server

5.13.2 Response Codes

5.13.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.13.2.2 Exception fault codes

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

5.13.3 GET

This operation is used to retrieve an outbound message request including the message delivery status.

5.13.3.1 Example (Informative)

5.13.3.1.1 Request

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId} HTTP/1.1
```

```
Accept: application/xml
Host: example.com:80
```

5.13.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:1350000001</address>
  <address>tel:1350000999</address>
  <senderAddress>tel:1351111999</senderAddress>
  <senderName>MyName</senderName>
  <!-- this is optional -->
  <receiptRequest>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </receiptRequest>
  <outboundMMSMessage>
    <subject>Holiday greetings</subject>
</outboundMMSMessage>
  <clientCorrelator>567895</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
  <deliveryInfoList>
      <!-- this is optional -->
     <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/DeliveryInfos
     </resourceURL>
    <deliveryInfo>
      <address>tel:1350000001</address>
      <deliveryStatus>MessageWaiting</deliveryStatus>
    </deliveryInfo>
    <deliveryInfo>
      <address>tel:1350000999</address>
      <deliveryStatus>MessageWaiting</deliveryStatus>
    </deliveryInfo>
  </deliveryInfoList>
</mms:outboundMessageRequest>
```

5.13.4 PUT

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

5.13.5 POST

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

5.13.6 DELETE

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

5.14 Resource: Outbound message delivery status

The resource used is:

http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos

This resource is used to request outbound message delivery status.

5.14.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)
senderAddress	identifies client application. Typically SHORT CODE [REST_TS_COMMON], but could be a terminal address
requestId	outbound message request identifier generated by server

5.14.2 Response Codes

5.14.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.14.2.2 Exception fault codes

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

5.14.3 GET

This operation is used to retrieve outgoing message delivery status.

Note: ParlayX SOAP equivalent is GetMessageDeliveryStatus.

5.14.3.1 Example (Informative)

5.14.3.1.1 Request

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.14.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryInfoList xmlns:mms="urn:oma:xml:rest:messaging:1">
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos
   </resourceURL>
  <deliveryInfo>
     <address>tel:1350000001</address>
     <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
  <deliveryInfo>
     <address>tel:1350000999</address>
     <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
</mms:deliveryInfoList>
```

5.14.4 PUT

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

5.14.5 POST

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

5.14.6 DELETE

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

5.15 Resource: Outbound message delivery notification subscriptions

The resource used is: **http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/subscriptions**

This resource gives access to outbound subscriptions for a particular client.

5.15.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST

apiVersion	version of the ParlayREST API clients want to use
senderAddress	identifies client application. Typically SHORT CODE [REST_TS_COMMON], but could be a terminal address.

5.15.2 Response Codes

5.15.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.15.2.2 Exception fault codes

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

5.15.3 GET

This operation is used to read all outbound message delivery notification subscriptions for the particular client.

Note: no equivalent ParlayX SOAP API.

5.15.3.1 Example (Informative)

5.15.3.1.1 Request

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.15.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryReceiptSubscriptionList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions</resourceURL>
  <deliveryReceiptSubscription>
   <callbackReference>
     <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
     <callbackData>12345</callbackData>
   </callbackReference>
   <filterCriteria>0102</filterCriteria>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}</resourceURL>
  </deliveryReceiptSubscription>
  <deliveryReceiptSubscription>
   <callbackReference>
     <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
     <callbackData>54321</callbackData>
   </callbackReference>
   <filterCriteria>0103</filterCriteria>
   <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}</resourceURL>
  </deliveryReceiptSubscription>
</mms:deliveryReceiptSubscriptionList>
```

5.15.4 PUT

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

5.15.5 POST

This operation is used to create a new outbound message delivery notification subscription for the particular client.

ParlayX SOAP equivalent is StartDeliveryReceiptNotification.

5.15.5.1 Example (Informative)

5.15.5.1.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryReceiptSubscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/77777</notifyURL>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
</mms:deliveryReceiptSubscription>
```

5.15.5.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: ../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId}

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryReceiptSubscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
   <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/77777</notifyURL>
  </callbackReference>
<filterCriteria>0102</filterCriteria>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId}</resourceURL>
</mms:deliveryReceiptSubscription>
```

5.15.6 DELETE

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

5.16 Resource: Individual outbound message delivery notification subscription

The resource used is:

http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId}

This resource controls individual subscription for outbound message delivery notification and gives access to individual subscription for a particular client.

5.16.1 Request URI variables

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

Name	Description
serverRoot	server base url: hostname+port+base path. Example: http://example.com:80/ParlayREST
apiVersion	version of the ParlayREST API client wants to use (e.g. version 1 for 1.x)
senderAddress	sender application address. Typically SHORT CODE [REST_TS_COMMON], but could be a terminal address.
subscriptionId	identifier of the subscription.

5.16.2 Response Codes

5.16.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.16.2.2 Exception fault codes

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

5.16.3 GET

This operation is used to read an individual outbound message delivery notification subscription for the particular client.

Note: no equivalent ParlayX SOAP API.

5.16.3.1 Example (Informative)

5.16.3.1.1 Request

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/xml
```

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

5.16.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryReceiptSubscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}</resourceURL>
</mms:deliveryReceiptSubscription>
```

5.16.4 PUT

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

5.16.5 POST

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

5.16.6 DELETE

This operation is used to delete a subscription for the particular client.

Note: ParlayX SOAP equivalent is StopDeliveryReceiptNotification.

5.16.6.1 Example (Informative)

5.16.6.1.1 Request

```
DELETE .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions /{subscriptionId} HTTP/1.1
Accept: application/xml
Host: example.com:80
```

5.16.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

5.17 Resource: Client notification about outbound message delivery status

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

5.17.1 Request URI variables

Client provided.

5.17.2 Response Codes

5.17.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.17.3 GET

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

5.17.4 PUT

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

5.17.5 POST

This operation is used to notify the client about outbound message delivery status

Note: ParlayX SOAP equivalent is NotifyMessageDeliveryReceipt.

5.17.5.1 Example 1: multiple delivery status per notification (Informative)

5.17.5.1.1 Request

```
POST …/notifications}/DeliveryInfoNotification/77777 HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: application.example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryInfoNotification xmlns:mms="urn:oma:xml:rest:messaging:1">
  <deliveryInfo>
    <address>tel:1350000001</address>
    <deliveryStatus>DeliveredToTerminal</deliveryStatus>
  </deliveryInfo>
  <deliveryInfo>
    <address>tel:1350000999</address>
    <deliveryStatus>DeliveredToTerminal</deliveryStatus>
</deliveryInfo>
  <link rel="OutboundMessageRequest"
  href="http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}" />
```

```
</mms:deliveryInfoNotification>
```

5.17.5.1.2 Response

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

5.17.5.2 Example 2: single delivery status per notification (Informative)

5.17.5.2.1 Request

```
POST …/notifications/DeliveryInfoNotification/77777 HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: application.example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<mms:deliveryInfoNotification xmlns:mms="urn:oma:xml:rest:messaging:1">
  <deliveryInfo>
    <address>tel:1350000999</address>
    <deliveryStatus>DeliveredToTerminal</deliveryStatus>
</deliveryInfo>
  <link rel="OutboundMessageRequest"
    href="http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}"/>
</mms:deliveryInfoNotification>
```

5.17.5.2.2 Response

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

5.17.6 DELETE

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

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-TS-ParlayREST-MultiMediaMessaging-V1_0	24 Jun 2009	All	Baseline TS uploaded as per agreed OMA-ARC-REST-2009-0009-INP_ParlayREST_MMS Editorial updates: versioning and history box fixed
	16 Nov 2009	4.1,5	Added after CC: OMA-ARC-REST-2009-0023-CR_ParlayREST_MultiMediaMessaging
	25 Nov 2009	4.1,5	Formatting and cleanup. Changes from OMA-ARC-REST-2009-0071R02-CR_Updates_to_MMS_TS and from OMA-ARC-REST-2009-0042R02-CR_ParlayREST_MMS_API-edits
	26 Nov 2009		Had to create a new version to upload the second part.
	1 Dec 2009	5.1	Added OMA-ARC-REST-2009-0076R01-CR_MMS_API_Optionality and OMA-ARC-REST-2009-0095R01-CR_Equivalent_PX_SOAP_to_MMS_TS and OMA-ARC-REST-2009-0070R01-CR_Vodafone_Comments_on_MMS_API and OMA-ARC-REST-2009-0091R01-CR_FlowDiagrams_to_MMS
	2 Des 2009	5.1, 5.2, 5.15, 5.16	Changes from OMA-ARC-REST-2009-0097-CR_Adding_Resources_to_MMS_TS.doc
	3 Dec 2009	4 2 3 5	Updated from OMA-ARC-REST-2009-0113-INP_SMS_Intro_section. Updated from OMA-ARC-REST-2009-0114-INP_SMS_TS_Reference_Section Updated from OMA-ARC-REST-2009-0115R01-INP_SMS_TS_Section_3. Update from OMA-ARC-REST-2009-0127R01-CR_Fix_retrievalOrder
	11 Dec 2009	all	Update after final CC, see OMA-ARC-REST-2009-0170-MINUTES_11Dec2009_CC for details OMA-ARC-REST-2009-0124R03-CR_Changes_to_MMS_API_alignment_with_98
	15 Dec 2009	all	Last corrections added: OMA-ARC-REST-2009-0148-CR_Issue_MMS_5_XML_examples OMA-ARC-REST-2009-0155-CR_SCR_and_PX_Profile_vs_REST_Ops_Appx_for_MMS OMA-ARC-REST-2009-0157-CR_Issue_MMS_17 OMA-ARC-REST-2009-0160R01-CR_MMS_urlencoded_examples_updated OMA-ARC-REST-2009-0163-CR_RequestError_Issue_MMS_2 OMA-ARC-REST-2009-0166R01-CR_SCR_MMS_TS
	16 Dec 2009	All	Editorial fixes: Styles as per template History Table
	26 Jan 2010	All	CONRR editorial comments applied, G005, G007, G008,
	03 Feb 2010	All	Applied G001, G002, G004
	04 Feb 2010	All	Added OMA-ARC-REST-2009-177 OMA-ARC-REST-2010-0023-CR_CONR_TS_Multimedia_Messaging_XML_Examples

Document Identifier	Date	Sections	Description
	05 Feb 2010	All	C001, C002,, C006, C005, C022, C045, C036, C032, C039 C008, C009, C010, C011, C012, C013, C017, C018, C019, C020, C021, C022, C023, C024, C025, C026, C027, C028, C029, C033, C034, C035 C037, C038, C040, C041, C043, C044, C045, C046, C047, C048, C049 C050, C051, C052, C053, C054, C055, C056, C057, C058, C059, C060 C061, C062, C063, C064, C065, C066, C069, C070, C071, C072, C073 C074, C075, C077, C078, C079,
	24 Feb 2010	Many	CRs implemented: OMA-ARC-REST-2009-0177 (fixed missing cahnge in section 2.2) OMA-ARC-REST-2010-0037 (added missing references to SVC and POL exceptions) OMA-ARC-REST-2010-0042 (fixed missing change in section 2.2) OMA-ARC-REST-2010-0053R01 OMA-ARC-REST-2010-0008R03 OMA-ARC-REST-2010-0009R02 OMA-ARC-REST-2010-0010R02 (closing C0040, C0035, C0064) CONRR comments: C001 (completed inbound part for IM) C0019, B004 (SMS comment to remove “online”/”offline” is applicable to MMS also, agreed by e-mail mandate on Feb 24 2010). Other changes: “DeliveryInfoList” replaced with “deliveryInfos” in resouces, as agreed during CONRR discussions. Subheadnigs for Example sections re-arranged as agreed during CONRR discussions.
	26 Feb 2010	Many	All XML examples replaced with validated one. Also some small changes forgotten in the previous version.
	09 Mar 2010	Many	CRs implemented: OMA-ARC-REST-2010-0110 OMA-ARC-REST-2010-0107R01 OMA-ARC-REST-2010-0090 OMA-ARC-REST-2010-0088R02 OMA-ARC-REST-2010-0085 OMA-ARC-REST-2010-0111R01-INP (the table in chapter 5.1 has been split and the header of the table is repeated for each page)
	17 Mar 2010	Many	Implemented all CRs that were agreed in REST R&A on March 10 OMA-ARC-REST-2010-0094 OMA-ARC-REST-2010-0102 OMA-ARC-REST-2010-0104R01 OMA-ARC-REST-2010-0106 Other CRs: OMA-ARC-REST-2010-74 Editorial fixes after a walk-through. **Note that this revision contains changes that require update of XSDs**
	25 Mar 2010	Many	Implemented agreed CRs: OMA-ARC-REST-2010-0125R01 OMA-ARC-REST-2010-0127 OMA-ARC-REST-2010-0128 OMA-ARC-REST-2010-0129 OMA-ARC-REST-2010-0133 OMA-ARC-REST-2010-0136R01
	30 Mar 2010	All	Implemented agreed CRs: OMA-ARC-REST-2010-0143R02 OMA-ARC-REST-2010-0150 OMA-ARC-REST-2010-0153 OMA-ARC-REST-2010-0156 Editorial fixes and 2010 copyright
Candidate Version: OMA-TS-ParlayREST_MultiMediaMessaging-V1_0	27 Apr 2010	All	Status changed to Candidate by TP: OMA-TP-2010-0186-INP_ParlayREST_V1_0_ERP_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for ParlayREST.MMS Server

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

B.1.1 SCR for ParlayREST.MMS.Inbound.Registration Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-OFF-S-001-M	Support for reliable inbound messages delivery	5.4	
PARLAYREST-MMS-INB-OFF-S-002-M	Retrieve messages from server - GET	5.4.3 C2	

B.1.2 SCR for ParlayREST.MMS.Inbound.Registration.RetrieveDelete Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-OFF-RETDEL-S-001-O	Support for inbound message delivery and delete	5.5	PARLAYREST-MMS-INB-OFF-RETDEL-S-002-O
PARLAYREST-MMS-INB-OFF- RETDEL-S-002-O	Retrieve and delete messages from server - POST	5.5.5	

B.1.3 SCR for ParlayREST.MMS.Individual.Inbound.Registration.RetrieveDelete Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-MIME-INB-OFF-RETDEL-S-001-O	Support for inbound message delivery and delete	5.6	PARLAYREST-MMS-MIME-INB-OFF- RETDEL-S-002-O
PARLAYREST-MMS-MIME-INB-OFF-RETDEL-S-002-O	Retrieve and delete one message from server - POST	5.6.4	

B.1.4 SCR for ParlayREST.MMS.Individual.Inbound Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-IND-INB-S-001-M	Support for inbound individual message delivery	5.7	
PARLAYREST-MMS-IND-INB-S-002-O	Retrieve one message from server - GET	5.7.3	
PARLAYREST-MMS-IND-INB-S-003-M	Confirm and delete retrieved message from server - DELETE	5.7.6	

B.1.5 SCR for ParlayREST.MMS.Attach.Individual.Inbound Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-ATTACH-IND-INB-S-001-O	Support for inbound individual message attachment delivery	5.8	PARLAYREST-MMS-ATTACH-IND-INB-S-002-O AND PARLAYREST-MMS-ATTACH-IND-INB-S-003-O
PARLAYREST-MMS-ATTACH-IND-INB-S-002-O	Retrieve one message attachment from server - GET	5.8.3	
PARLAYREST-MMS-ATTACH-IND-INB-S-003-O	Confirm and delete retrieved message attachment from server - DELETE	5.8.6	

B.1.6 SCR for ParlayREST.MMS.Inbound.Subscr Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-ONL-SUBSCR-S-001-M	Support inbound subscriptions	5.9	
PARLAYREST-MMS-INB-ONL-SUBSCR-S-002-O	Read active subscriptions - GET	5.9.3	
PARLAYREST-MMS-INB-ONL-SUBSCR-S-003-M	Create inbound message subscription - POST	5.9.5 C.4.1	

B.1.7 SCR for ParlayREST.MMS.Inbound.Individual.Subscr Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-INDON-SUBSCR-S-001-M	Support for control and read access to individual inbound subscription	5.10	
PARLAYREST-MMS-INB-INDON-S-002-O	Read individual inbound subscription - GET	5.10.3	

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-INDON-S-003-M	Update individual inbound subscriptions - DELETE	5.10.6	

B.1.8 SCR for ParlayREST.MMS.Inbound.Notifications Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-INB-NOTIF-S-001-M	Support for notifying application about inbound messages	5.11	
PARLAYREST-MMS-INB-NOTIF-S-002-M	Notify application about inbound message arrival - POST	5.11.5	

B.1.9 SCR for ParlayREST.MMS.Outbound Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-OUTB-S-001-M	Support for outbound messages	5.12	
PARLAYREST-MMS-OUTB-S-002-O	Retrieve list of pending outgoing message requests - GET	5.12.3	
PARLAYREST-MMS-OUTB-S-003-M	Create outgoing message request - POST	5.12.5 C.1	

B.1.10 SCR for ParlayREST.MMS.Outbound.MsgAndDeliveryStatus Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-OUTB-MSGDELSTAT-S-001-O	Support for requesting an outbound message and its delivery status	5.13	PARLAYREST-MMS-OUTB-MSGDELSTAT-S-002-O
PARLAYREST-MMS-OUTB-MSGDELSTAT-S-002-O	Retrieve Outgoing Message Delivery Status - GET	5.13.3	

B.1.11 SCR for ParlayREST.MMS.Outbound.DeliveryStatus Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-OUTB-DELSTAT-S-001-M	Support for requesting delivery status of outbound messages	5.14	
PARLAYREST-MMS-OUTB-DELSTAT-S-002-M	Retrieve Outgoing Message Delivery Status - GET	5.14.3	

B.1.12 SCR for ParlayREST.MMS.Outbound.Subscriptions Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-OUTB-SUBSCR-S-001-M	Support for outbound subscriptions for a particular client	5.15	
PARLAYREST-MMS-OUTB-SUBSCR-S-002-O	Read all outbound message delivery notification subscriptions - GET	5.15.3	
PARLAYREST-MMS-OUTB-SUBSCR-S-003-M	Create new outbound message subscription - POST	5.15.5 C.3.1	

B.1.13 SCR for ParlayREST.MMS.Individual.Outbound.Subscr Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-IND-OUTB-IND-SUBSCR-S-001-M	Support for outbound subscriptions for a particular client	5.16	
PARLAYREST-MMS-IND-OUTB-IND-SUBSCR-S-002-O	Read individual message delivery notification subscription - GET	5.16.3	
PARLAYREST-MMS-IND-OUTB-IND-SUBSCR-S-003-M	Delete subscription for the client - DELETE	5.16.6	

B.1.14 SCR for ParlayREST.MMS.Outbound.DeliveryStatus.Notifications Server

Item	Function	Reference	Requirement
PARLAYREST-MMS-OUTB-DELSTAT-NOTIF-S-001-M	Support for notifying application about delivery status of outbound messages	5.17	
PARLAYREST-MMS-OUTB-DELSTAT-NOTIF-S-002-M	Notify application about delivery status of outbound message - POST	5.17.5	

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

For selected operations, this section defines a format for MMS REST API requests where the body of the **request** is encoded using the application/*x-www-form-urlencoded* MIME type. XML wrapper elements (the root of a complexType) have been omitted from the *x-www-form-urlencoded* model, since it has no hierarchy. Instead the sub-elements of the complex Type are represented in the parameter tables below.

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 at [W3C-URLENC].

x-www-form-urlencoded bindings for the following MMS REST operations are defined in this section:

- Sending a message to a terminal
- A polling mechanism for monitoring the delivery status of a sent message
- A mechanism to start the notification of delivery receipts
- A polling mechanism to receive messages
- A mechanism to start the notification of received messages

C.1 Send a message to a terminal

This operation is used to create an outgoing message request.

Note: ParlayX SOAP equivalent is SendMessage for MMS.

The request parameters are as follows:

Name	Type/Values	Optional	Description
address	xsd:anyURI [1…unbounded]	No	Destination address(es) for the message
senderAddress	xsd:anyURI	No	User Identity of the Sender of the message. The associated MSISDN will appear in the receiver terminal, unless a senderName is specificated. This parameter shall match the User's Identity included in the Authorization header.
senderName	xsd:string	Yes	Name of the sender to appear on the user's terminal as the originator of the message. If this parameter is used, a set of allowed values shall be set during provisioning each sender (i.e.: for each User provisioned in the System).
chargingDescription	xsd:string [0..unbounded]	Yes	Description of charge to apply to this message. In case charging is required, this parameter MUST be present.
chargingCurrency	xsd:string	Yes	Currency of charge to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.

chargingAmount	xsd:decimal	Yes	Charging amount to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.
chargingCode	xsd:string	Yes	Charging code to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.
notifyURL	xsd:anyURI	Yes	URL to notify the application for delivery receipts
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}.
clientCorrelator	xsd:string	Yes	A correlator that the client SHOULD use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
subject	xsd:string	Yes	If present, indicates the subject of the received message.
priority	MessagePriority	Yes	The priority of the message: default is Normal.

C.1.1 Example (Informative)

C.1.1.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Length: nnnnnn
Content-Type: multipart/form-data;
            boundary="===============123456==";

MIME-Version: 1.0
Host: www.example.com
Date: Thu, 04 Jun 2009 02:51:59 GMT


--===============123456==
Content-Disposition: form-data; name="root-fields"
Content-Type: application/x-www-form-urlencoded;
address=tel:%2B13500000991&
```

```
address=tel:%2B13500000992&
senderAddress=tel:%2B12345678&
Subject=My%20message&
notifyURL=http://example-application.com/notifications/DeliveryInfoNotification/54311&
clientCorrelator=123456&senderName=Bob
--===============123456==
Content-Disposition: form-data; name=”attachments”; filename=”picture.jpg”
Content-Type: image/gif

GIF89a...binary image data...
--===============123456==--
```

C.1.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:outboundMessageRequest xmlns:mms="urn:oma:xml:rest:messaging:1">
  <address>tel:135000000991</address>
  <address>tel:13500000992</address>
  <senderAddress>tel:12345678</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/54311</notifyURL>
  </receiptRequest>
  <outboundMMSMessage>
    <subject>Holiday greetings</subject>
</outboundMMSMessage>
<clientCorrelator>567895</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
</mms:outboundMessageRequest>
```

C.2 Retrieve inbound messages received

This operation is used for reliable inbound message delivery for the particular client.

Note: ParlayX SOAP equivalent is GetReceivedMessage.

Request parameters are as follows:

Name	Type/Values	Optional	Description
maxBatchSize	xsd:int	Yes	Specifies maximum number of messages to be returned in the response

retrievalOrder	RetrievalOrder	Yes	Specifies order in which messages should be retrieved is there are more then one pending
priority	MessagePriority	Yes	The priority of the messages to poll from the gateway. All messages of the specified priority and higher will be retrieved. If not specified, all messages shall be returned, i.e. the same as specifying Low.
useAttachmentURLs	xsd:boolean	Yes	Default: false If set to 'true', inbound message would have links to attachments. Otherwise, only message identifier will be returned, so that individual message retrieval can be done.

If the operation was successful, this would return an HTTP Status: 200 OK and a list of the received messages since the last invocation.

C.2.1 Example (Informative)

C.2.1.1 Request

```
GET .../{apiVersion}/messaging/inbound/subscriptions/{registrationid}/messages?maxBatchSize=2&
useAttachmentURLs=true HTTP/1.1
Accept: application/xml
Host: www.example.com
```

C.2.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<mms:inboundMessageList xmlns:mms="urn:oma:xml:rest:messaging:1">
  <inboundMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}
    </resourceURL>
    <messageId>{messageId1}</messageId>
    <inboundMMSMessage>
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <!-- MMS  -->
  <inboundMessage>
    <destinationAddress>MSISDN3</destinationAddress>
    <senderAddress>MSISDN4</senderAddress>
    <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId2}
    </resourceURL>
    <messageId>{messageId2}</messageId>
    <inboundMMSMessage>
```

```
      <subject>Who is RESTing on the beach?</subject>
    </inboundMMSMessage>
  </inboundMessage>
  <totalNumberOfPendingMessages>20</totalNumberOfPendingMessages>
  <numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
  <resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages</resourceURL>
</mms:inboundMessageList>
```

C.3 Start delivery receipt notification

This REST method is used by the application to subscribe for the delivery receipt notifications. It MUST use the HTTP POST method. It is equivalent of the Parlay X StartDeliveryReceiptNotification.

Request parameters are as follows:

Name	Type/Values	Optional	Description
filterCriteria	xsd:string	No	The FilterCriteria will allow the service to filter flexibly. One example would be for the Service Provider to filter based on first 4 digits in MSISDN. This however is implementation specific and will be left to the Service Provider.
notifyURL	xsd:anyURI	No	Notification endpoint definition
callbackData	xsd:string	No	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}.
clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.

C.3.1 Example (Informative)

C.3.1.1 Request

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions HTTP/1.1
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
Accept: application/xml
```

```
filterCriteria=13500&
correlator=123456&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification/12345
```

C.3.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Date: Thu, 04 Jun 2009 02:51:59 GMT
Location: ../{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}

<?xml version="1.0" encoding="UTF-8"?>
<mms: deliveryReceiptSubscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/12345</notifyURL>
  </callbackReference>
<filterCriteria>0102</filterCriteria>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId}
   </resourceURL>
</mms:deliveryReceiptSubscription>
```

C.4 Start message notification

This REST method is used by the application to subscribe for the notifications of received messages. It MUST use the HTTP POST method. It is the equivalent of the Parlay X StartMessageNotification.

Request parameters are as follows:

Name	Type/Values	Optional	Description
destinationAddress	xsd:anyURI[1..unbounded]	No	Destination address of the message
criteria	xsd:string	Yes	The text to match against to determine the application to receive the notification
notifyURL	xsd:anyURI	No	Notification endpoint definition
callbackData	xsd:string	No	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}.

clientCorrelator	xsd:string	Yes	A correlator that the client MAY use to tag this particular resource representation during a request to create a resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
useAttachmentURLs	xsd:boolean	Yes	Default: false If set to 'true', inbound message would have links to attachments. Otherwise, only message identifier will be returned, so that individual message retrieval can be done

This operation would return a result indicating whether the operation has been successful.

C.4.1 Example (Informative)

C.4.1.1 Request

```
POST .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Host: www.example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/xml

destinationAddress=81771&
criteria=Vote&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification/3455&
notificationFormat=XML
```

C.4.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

?xml version="1.0" encoding="UTF-8"?>
<mms:subscription xmlns:mms="urn:oma:xml:rest:messaging:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/3455</notifyURL>
    <notificationFormat>XML</notificationFormat>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
<criteria>Urgent*</criteria>
<resourceURL>http://{serverRoot}/{apiVersion}/messaging/inbound/subscription/{subscriptionId1}</resourceURL>
  <useAttachmentURLs>false</useAttachmentURLs>
```

```
</mms:subscription>
```

Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for Parlay REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC4627].

The following examples show the request or response for various operations using a JSON binding. The examples follow the XML to JSON serialization guidelines in [REST_WP]. A JSON respone may be obtained by following the content negotiation guidelines section of [REST_WP].

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

D.1 Retrieve messages for a registration (section 5.4.3.1)

Request:

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2 HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"inboundMessageList": {
    "inboundMessage": [
        {
            "destinationAddress": "MSISDN1",
            "messageId": "{messageId1}",
            "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
            "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}",
            "senderAddress": "MSISDN2"
        },
        {
            "destinationAddress": "MSISDN3",
            "messageId": "{messageId2}",
            "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
            "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId2}",
            "senderAddress": "MSISDN4"
        }
    ],
    "numberOfMessagesInThisBatch": "2",
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2",
    "totalNumberOfPendingMessages": "20"
}}
```

D.2 Request with invalid id (section 5.4.3.2)

Request:

```
GET .../{apiVersion}/messaging/inbound/registrations/registration123/messages?maxBatchSize=2 HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"requestError": {
  "link": {
    "href": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/registration123/messages?maxBatchSize=2",
    "rel": "self"
  },
  "serviceException": {
    "messageId": "SVC0002",
    "text": "Invalid input value. The requested registration id: %1 is not valid ",
    "variables": "registration123"
  }
}}
```

D.3 Retrieve messages with attachment URLs (section 5.4.3.3)

Request:

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2&useAttachmentURLs=true HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"inboundMessageList": {
  "inboundMessage": {
    "destinationAddress": "MSISDN1",
    "messageId": "{messageId1}",
    "inboundMMSMessage": {
      "bodyText": "See attached picture",
      "link": [
        {
          "href":
"http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{mesageId1}/attachments/{attachmentId1}",
          "rel": "attachment"
        },
        {
          "href":
"http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}/attachments/{attachmentId2}",
          "rel": "attachment"
```

```
                }
            ],
            "subject": "Who is RESTing on the beach?"
        },
        "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}",
        "senderAddress": "MSISDN2"
    },
    "numberOfMessagesInThisBatch": "2",
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}/attachments",
    "totalNumberOfPendingMessages": "20"
}}
```

D.4 Retrieve and delete inbound messages (section 5.5.5.1)

Request:

```
POST .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages  HTTP/1.1
Accept: application/json
Content-Length: nnn
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"inboundMessageRetrieveAndDeleteRequest": {"retrievalOrder": "OldestFirst"}}
```

Response:

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

{"inboundMessageList": {
    "inboundMessage": [
        {
            "destinationAddress": "MSISDN1",
            "messageId": "{messageId1}",
            "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
            "senderAddress": "MSISDN2"
        },
        {
            "destinationAddress": "MSISDN3",
            "messageId": "{messageId2}",
            "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
            "senderAddress": "MSISDN4"
        }
    ],
    "numberOfMessagesInThisBatch": "2",
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/retrieveAndDeleteMessages",
    "totalNumberOfPendingMessages": "20"
}}
```

D.5 Read and delete one message (section 5.6.4)

Request:

```
POST .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/retrieveAndDelete HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Length: nnnnnn
Content-Type: multipart/form-data;
                boundary="===============123456==";
                type="application/json"
MIME-Version: 1.0
Date: Thu, 04 Jun 2009 02:51:59 GMT

--===============123456==
Content-Disposition: form-data; name=”root-fields”
Content-Type: application/json

{"inboundMessage": {
   "destinationAddress": "MSISDN1",
   "messageId": "{messageId1}",
   "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}",
   "senderAddress": "MSISDN2"
}}
```

D.6 Read message from gateway storage (section 5.7.3.1)

Request:

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }?resFormat=JSON HTTP/1.1
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Type: multipart/form-data; boundary=”=====12345====”
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

====12345====
Content-Disposition=multipart/form-data; name=”root-fields”
Content-Type=application/json
Content-Length: nnnn
{"inboundMessage": {
   "destinationAddress": "MSISDN1",
   "messageId": "{messageId1}",
   "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}",
   "senderAddress": "MSISDN2"
}}
```

D.7 Remove message from gateway storage (section 5.7.6)

Request:

```
DELETE .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId } HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.8 Read an MMS attachment (section 5.8.3)

Request:

```
GET .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId} HTTP/1.1
Accept: image/gif, image/png, image/jpeg, text/html, application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
Content-Length: nnnnnn
Content-Type: image/gif
Date: Thu, 04 Jun 2009 02:51:59 GMT

...GIF89a...binary image data
```

D.9 Delete an MMS attachment from gateway storage (section 5.8.6)

Request:

```
DELETE .../{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId }/attachments/{attatchmentId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.10 Read active subscriptions (section 5.9.3)

Request:

```
GET .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"subscriptionList": {"subscription": [
    {
        "callbackReference": {
            "callbackData": "12345",
            "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/12345"
        },
        "clientCorrelator": "567891",
        "criteria": "Urgent*",
        "destinationAddress": "680180999",
        "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/0000001",
        "useAttachmentURLs": "false"
    },
    {
        "callbackReference": {
            "callbackData": "54321",
            "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/54321"
        },
        "clientCorrelator": "567892",
        "criteria": "Urgent*",
        "destinationAddress": "80999",
        "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/0000002",
        "useAttachmentURLs": "false"
    }
]}}
```

D.11 Create inbound subscription (section 5.9.5)

Request:

```
POST .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"subscription": {
    "callbackReference": {
        "callbackData": "12345",
        "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/88888"
    },
    "clientCorrelator": "567893",
    "criteria": "Urgent*",
    "destinationAddress": "+34680180999",
    "useAttachmentURLs": "false"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}
Content-Length: 254
```

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

{"subscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/88888"
   },
   "clientCorrelator": "567893",
   "criteria": "Urgent*",
   "destinationAddress": "+34680180999",
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions{subscriptionId1}",
   "useAttachmentURLs": "false"
}}
```

D.12 Returning location of created resource (section 5.9.5.2)

Request:

```
POST .../{apiVersion}/messaging/inbound/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"subscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
   },
   "criteria": "Urgent*",
   "destinationAddress": "+34680180999",
   "useAttachmentURLs": "false"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}"}}
```

D.13 Read individual subscription (section 5.10.3)

Request:

```
GET .../{apiVersion}/messaging/inbound/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 200 OK
```

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

{"subscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/88888"
   },
   "clientCorrelator": "567893",
   "criteria": "Urgent*",
   "destinationAddress": "+34680180999",
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId}",
   "useAttachmentURLs": "false"
}}
```

D.14 Delete a subscription (section 5.10.6)

Request:

```
DELETE .../{apiVersion}/messaging/inbound/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.15 Message arrival notification (section 5.11.5.1.1)

Request:

```
POST …/notifications/DeliveryInfoNotification/88888 HTTP/1.1
Accept: application/json
Content-Length: 12345
Host: application.example.com:80
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"inboundMessageNotification": {"inboundMessage": {
   "destinationAddress": "MSISDN1",
   "messageId": "{messageId1}",
   "inboundMMSMessage": {"subject": "Who is RESTing on the beach?"},
   "link": {
      "href": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}",
      "rel": "Subscription"
   },
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}",
   "senderAddress": "MSISDN2"
}}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.16 Message arrival notification with attachment URLs (section 5.11.5.2)

Request:

```
POST http://example-application.com/notifications/DeliveryInfoNotification HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
Host: application.example.com:80

{"inboundMessageNotification": {"inboundMessage": {
    "destinationAddress": "MSISDN1",
    "messageId": "{messageId1}",
    "inboundMMSMessage": {
        "bodyText": "Look at the attached picture",
        "link": [
            {
                "href": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}/attachments/{attachment
id1}",
                "rel": "attachment"
            },
            {
                "href": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/{messageId1}/attachments/{attachment
id2}",
                "rel": "attachment"
            }
        ],
        "subject": "Who is RESTing on the beach?"
    },
    "link": {
        "href": "http://{serverRoot}/{apiVersion}/messaging/inbound/subscriptions/{subscriptionId1}",
        "rel": "Subscription"
    },
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/inbound/registrations/{registrationId}/messages/{messageId1}",
    "senderAddress": "MSISDN2"
}}}
```

Response:

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

D.17 Retrieve list of outgoing requests (section 5.12.3)

Request:

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"outboundMessageRequestList": {"outboundMessageRequest": {
  "address": "tel:1350000001",
  "clientCorrelator": "567894",
  "deliveryInfoList": {
    "deliveryInfo": {
      "address": "tel:1350000001",
      "deliveryStatus": "DeliveredToTerminal"
    },
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId1}/deliveryInfos"
  },
  "outboundMMSMessage": {"subject": "Holiday greetings"},
  "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId1}",
  "senderAddress": "tel:1350000009"
}}}
```

D.18 Create outgoing message (section 5.12.5.1)

Request:

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/form-data; boundary="===============123456==";
Host: example.com:80
MIME-Version: 1.0

--===============123456==
Content-Disposition: multipart/form-data; name=”root-fields”
Content-Type: application/json; charset="utf-8"
Content-Length: nnnn

{"outboundMessageRequest": {
  "address": [
    "tel:1350000001",
    "tel:1350000999"
  ],
  "clientCorrelator": "567895",
  "outboundMMSMessage": {
    "priority": "High",
    "subject": "hello from the rest of us!"
  },
  "receiptRequest": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
  "senderAddress": "tel:1351111999",
  "senderName": "MyName"
}}
```

Response:

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

```
Location: http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundMessageRequest": {
   "address": [
      "tel:1350000001",
      "tel:1350000999"
   ],
   "clientCorrelator": "567895",
   "outboundMMSMessage": {"subject": "Holiday greeings"},
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}",
   "senderAddress": "tel:1351111999",
   "senderName": "MyName"
}}
```

D.19 Create message returning resource location (section 5.12.5.2.1)

Request:

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/form-data; boundary="===============123456==";
Host: example.com:80
MIME-Version: 1.0

--===============123456==
Content-Disposition: multipart/form-data; name=”root-fields”
Content-Type: application/json; charset="utf-8"
Content-Length: nnnn

{"outboundMessageRequest": {
   "address": [
      "tel:1350000001",
      "tel:1350000999"
   ],
   "clientCorrelator": "567895",
   "outboundMMSMessage": {
      "priority": "High",
      "subject": "hello from the rest of us!"
   },
   "receiptRequest": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
   },
   "senderAddress": "tel:1351111999",
   "senderName": "MyName"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}
```

```
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": " http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}"}}
```

D.20 Create message with charging (section 5.12.5.3.1)

Request:

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/requests HTTP/1.1
Content-Type: multipart/related; boundary="===============123456==";type="application/json"
Host: example.com:80
MIME-Version: 1.0

--===============123456==
Content-Type: application/json; charset="utf-8"

{"outboundMessageRequest": {
    "address": [
        "tel:1350000001",
        "tel:1350000999"
    ],
    "charging": {"description": "Sample text for the charging information"},
    "clientCorrelator": "567896",
    "outboundMMSMessage": {
        "priority": "High",
        "subject": "hello from the rest of us!"
    },
    "receiptRequest": {
        "callbackData": "12345",
        "notifyURL": "http://example-application.com/notifications/DeliveryInfoNotification"
    },
    "senderAddress": " tel:1351111999",
    "senderName": "MyName"
}}
```

Response for charging not supported:

```
HTTP/1.1 400 Bad request
Content-Type: application/json
Content-Length: 254
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {"policyException": {
    "messageId": "POL0008",
    "text": "Charging is not supported"
}}}
```

D.21 Read message request and delivery status (section 5.13.3)

Request:

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId} HTTP/1.1
Accept: application/json
```

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

Response:

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

{"outboundMessageRequest": {
   "address": [
      "tel:1350000001",
      "tel:1350000999"
   ],
   "clientCorrelator": "567895",
   "deliveryInfoList": {
      "deliveryInfo": [
         {
            "address": "tel:1350000001",
            "deliveryStatus": "MessageWaiting"
         },
         {
            "address": "tel:1350000999",
            "deliveryStatus": "MessageWaiting"
         }
      ],
      "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/DeliveryInfos"
   },
   "outboundMMSMessage": {"subject": "Holiday greetings"},
   "receiptRequest": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
   },
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}",
   "senderAddress": "tel:1351111999",
   "senderName": "MyName"
}}
```

D.22 Read message delivery status (section 5.14.3)

Request:

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"deliveryInfoList": {
   "deliveryInfo": [
        {
            "address": "tel:1350000001",
            "deliveryStatus": "MessageWaiting"
        },
        {
            "address": "tel:1350000999",
            "deliveryStatus": "MessageWaiting"
        }
    ],
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/DeliveryInfo"
}}
```

D.23 Read delivery notification subscriptions (section 5-15-3-1)

Request:

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"deliveryReceiptSubscriptionList": {
    "deliveryReceiptSubscription": [
        {
            "callbackReference": {
                "callbackData": "12345",
                "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
            },
            "filterCriteria": "0102",
            "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}"
        },
        {
            "callbackReference": {
                "callbackData": "54321",
                "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
            },
            "filterCriteria": "0103",
            "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}"
        }
    ],
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions"
}}
```

D.24 Create outbound delivery notification subscription (section 5.15.5)

Request:

```
POST .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/campaign/notifications/DeliveryInfoNotification/77777"
   },
   "filterCriteria": "0102"
}}
```

Response:

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

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/77777"
   },
   "filterCriteria": "0102",
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId}"
}}
```

D.25 Read individual message delivery notification subscription (section 5.16.3)

Request:

```
GET .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions/{subscriptionId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

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

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
   },
   "filterCriteria": "0102",
   "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/subscriptions/{subscriptionId}"
}}
```

D.26 Delete message delivery notification subscription (section 5.16.6.1.1)

Request:

```
DELETE .../{apiVersion}/messaging/outbound/{senderAddress}/subscriptions /{subscriptionId} HTTP/1.1
Accept: application/json
Host: example.com:80
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

D.27 Notify client about outbound message delivery status (section 5.17.5)

Request:

```
POST http://example-application.com/notifications}/DeliveryInfoNotification/77777 HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: application.example.com:80

{"deliveryInfoList": {
    "deliveryInfo": [
        {
            "address": "tel:1350000001",
            "deliveryStatus": "DeliveredToTerminal"
        },
        {
            "address": "tel:1350000999",
            "deliveryStatus": "DeliveredToTerminal"
        }
    ],
    "link": {
        "href": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId)",
        "rel": "OutboundMessageRequest"
    },
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/DeliveryInfos"
}}
```

Response:

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

D.28 Single delivery status per notification (section 5.17.5.2)

Request:

```
POST .../notifications/DeliveryInfoNotification/77777 HTTP/1.1
```

```
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80
{"deliveryInfoList": {
    "deliveryInfo": {
        "address": "tel:1350000999",
        "deliveryStatus": "DeliveredToTerminal"
    },
    "link": {
        "href": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}",
        "rel": "self"
    },
    "resourceURL": "http://{serverRoot}/{apiVersion}/messaging/outbound/{senderAddress}/requests/{requestId}/DeliveryInfos"
}}
```

Response:

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