RESTful bindings for Parlay X Web Services –
Short Messaging
Candidate Version 1.0 – 23 Nov 2010

Open Mobile Alliance
OMA-TS-ParlayREST_ShortMessaging-V1_0-20101123-C
## Contents

1. **SCOPE** ......................................................................................................................................................................... 8

2. **REFERENCES** .................................................................................................................................................................. 9
   2.1 NORMATIVE REFERENCES ........................................................................................................................................... 9
   2.2 INFORMATIVE REFERENCES ........................................................................................................................................ 9

3. **TERMINOLOGY AND CONVENTIONS** ................................................................................................................................. 10
   3.1 CONVENTIONS ............................................................................................................................................................... 10
   3.2 DEFINITIONS ................................................................................................................................................................. 10
   3.3 ABBREVIATIONS ............................................................................................................................................................ 10

4. **INTRODUCTION** .............................................................................................................................................................. 11
   4.1 VERSION 1.0 .................................................................................................................................................................... 11

5. **SHORT MESSAGING SERVICE (SMS) API DEFINITION** .................................................................................................. 12
   5.1 RESOURCES SUMMARY .................................................................................................................................................. 12
   5.2 SMS PARLAYREST API DATA STRUCTURES ...................................................................................................................... 17
   5.2.1 Type: InboundSMSMessageList ................................................................................................................................ 18
   5.2.2 Type: InboundSMSMessage .................................................................................................................................... 18
   5.2.3 Type: InboundSMSMessageNotification ................................................................................................................... 18
   5.2.4 Type: SubscriptionList ............................................................................................................................................. 19
   5.2.5 Type: Subscription .................................................................................................................................................. 19
   5.2.6 Type: InboundSMSMessageRetrieveAndDeleteRequest .......................................................................................... 20
   5.2.7 Type: OutboundSMSMessageRequestList .................................................................................................................... 20
   5.2.8 Type: OutboundSMSMessageRequest ........................................................................................................................ 21
   5.2.9 Type: OutboundSMSTextMessage ............................................................................................................................. 22
   5.2.10 Type: OutboundSMSBinaryMessage ........................................................................................................................ 22
   5.2.11 Type: OutboundSMSLogoMessage .......................................................................................................................... 23
   5.2.12 Type: OutboundSMSSingleMessage .......................................................................................................................... 23
   5.2.13 Type: DeliveryInfoList ........................................................................................................................................... 23
   5.2.14 Type: DeliveryInfoNotification ................................................................................................................................ 23
   5.2.15 Type: DeliveryInfo ................................................................................................................................................ 24
   5.2.16 Type: DeliveryReceiptSubscriptionList .................................................................................................................. 24
   5.2.17 Type: DeliveryReceiptSubscription ........................................................................................................................ 25
   5.2.18 Enumeration: DeliveryStatus ................................................................................................................................... 25
   5.2.19 Enumeration: SmsFormat ......................................................................................................................................... 26
   5.2.20 Enumeration: RetrievalOrder .................................................................................................................................. 26
   5.2.21 Values of the Link “rel” attribute ............................................................................................................................ 26

5.3 SEQUENCE DIAGRAMS ....................................................................................................................................................... 27
   5.3.1 Send SMS and check the delivery status .................................................................................................................. 27
   5.3.2 Inbound SMS message delivery (push mode) ........................................................................................................... 28
   5.3.3 Inbound SMS message delivery (polling mode) ......................................................................................................... 29

5.4 RESOURCE: INBOUND SMS MESSAGE REQUESTS FOR A GIVEN REGISTRATION ................................................................ 30
   5.4.1 Request URI variables .................................................................................................................................................. 30
   5.4.2 Response Codes ............................................................................................................................................................ 30
   5.4.2.1 Response Codes ..................................................................................................................................................... 30
   5.4.2.2 Exception fault codes ............................................................................................................................................ 30
   5.4.3 GET ............................................................................................................................................................................ 30
   5.4.3.1 Example 1: regular request (Informative) .................................................................................................................... 30
     5.4.3.1.1 Request ......................................................................................................................................................... 30
     5.4.3.1.2 Response ....................................................................................................................................................... 31
   5.4.3.2 Example 2: maxBatchSize exceeding the allowed size (Informative) .................................................................... 31
     5.4.3.2.1 Request ......................................................................................................................................................... 31
     5.4.3.2.2 Response ....................................................................................................................................................... 31

5.4.4 PUT ............................................................................................................................................................................. 32

5.4.5 POST ........................................................................................................................................................................... 32

5.4.6 DELETE ....................................................................................................................................................................... 32
5.5 **RESOURCE: INBOUND SMS MESSAGES RETRIEVE AND DELETE USING REGISTRATION** ......................................................... 32
5.5.1 Request URI variables ...................................................................................................................................................... 32
5.5.2 Response Codes ............................................................................................................................................................... 33
5.5.2.1 Response Codes ...................................................................................................................................................... 33
5.5.2.2 Exception fault codes ............................................................................................................................................... 33
5.5.3 GET .................................................................................................................................................................................. 33
5.5.4 PUT .................................................................................................................................................................................. 33
5.5.5 POST ................................................................................................................................................................................. 33
5.5.5.1 Example (Informative) ........................................................................................................................................... 33
5.5.5.1.1 Request ......................................................................................................................................................... 33
5.5.5.1.2 Response ..................................................................................................................................................... 34
5.5.6 DELETE .......................................................................................................................................................................... 34

5.6 **RESOURCE: INBOUND SMS MESSAGE FOR A GIVEN REGISTRATION** ................................................................................. 34
5.6.1 Request URI variables ...................................................................................................................................................... 34
5.6.2 Response Codes ............................................................................................................................................................... 35
5.6.2.1 HTTP Response Codes ........................................................................................................................................... 35
5.6.3 GET .................................................................................................................................................................................. 35
5.6.3.1 Example 1: regular request (Informative) .............................................................................................................. 35
5.6.3.1.1 Request ......................................................................................................................................................... 35
5.6.3.1.2 Response ..................................................................................................................................................... 35
5.6.3.2 Example 2: invalid (non-existing) messageId (Informative) .................................................................................. 35
5.6.3.2.1 Request ......................................................................................................................................................... 35
5.6.3.2.2 Response ..................................................................................................................................................... 36
5.6.4 PUT .................................................................................................................................................................................. 36
5.6.5 POST ................................................................................................................................................................................. 36
5.6.6 DELETE .......................................................................................................................................................................... 36
5.6.6.1 Example (Informative) ........................................................................................................................................... 36
5.6.6.1.1 Request ......................................................................................................................................................... 36
5.6.6.1.2 Response ..................................................................................................................................................... 36

5.7 **RESOURCE: INBOUND SMS MESSAGE SUBSCRIPTIONS** .................................................................................................. 36
5.7.1 Request URI variables ...................................................................................................................................................... 37
5.7.2 Response Codes ............................................................................................................................................................... 37
5.7.2.1 HTTP Response Codes ........................................................................................................................................... 37
5.7.2.2 Exception fault codes ............................................................................................................................................... 37
5.7.3 GET .................................................................................................................................................................................. 37
5.7.3.1 Example (Informative) ........................................................................................................................................... 37
5.7.3.1.1 Request ......................................................................................................................................................... 37
5.7.3.1.2 Response ..................................................................................................................................................... 37
5.7.4 PUT .................................................................................................................................................................................. 38
5.7.5 POST ................................................................................................................................................................................. 38
5.7.5.1 Example 1: returning a representation of created resource (Informative) ................................................................. 38
5.7.5.1.1 Request ......................................................................................................................................................... 38
5.7.5.1.2 Response ..................................................................................................................................................... 38
5.7.5.2 Example 2: returning the location of created resource (Informative) ...................................................................... 39
5.7.5.2.1 Request ......................................................................................................................................................... 39
5.7.5.2.2 Response ..................................................................................................................................................... 39
5.7.6 DELETE .......................................................................................................................................................................... 39

5.8 **RESOURCE: INDIVIDUAL INBOUND SMS MESSAGE SUBSCRIPTION** .................................................................................. 39
5.8.1 Request URI variables ...................................................................................................................................................... 40
5.8.2 Response Codes ............................................................................................................................................................... 40
5.8.2.1 HTTP Response Codes ........................................................................................................................................... 40
5.8.2.2 Exception fault codes ............................................................................................................................................... 40
5.8.3 GET .................................................................................................................................................................................. 40
5.8.3.1 Example (Informative) ........................................................................................................................................... 40
5.8.3.1.1 Request ......................................................................................................................................................... 40
5.8.3.1.2 Response ..................................................................................................................................................... 40
5.8.4 PUT .................................................................................................................................................................................. 41
5.8.5 POST ................................................................................................................................................................................. 41
5.8.6 DELETE .......................................................................................................................................................................... 41
5.8.6.1 Example (Informative) ........................................................................................................................................... 41
5.9 RESOURCE: CLIENT NOTIFICATION ABOUT INBOUND SMS MESSAGE ................................................................. 41
5.9.1 Request URI variables ........................................................................................................................................ 41
5.9.2 HTTP Response Codes .................................................................................................................................... 41
5.9.3 GET .................................................................................................................................................................. 41
5.9.4 PUT ................................................................................................................................................................. 41
5.9.5 POST ............................................................................................................................................................... 42
5.9.5.1 Example (Informative) ................................................................................................................................ 42
5.9.5.1.1 Request .................................................................................................................................................... 42
5.9.5.1.2 Response ................................................................................................................................................ 42
5.9.6 DELETE ......................................................................................................................................................... 42
5.10 RESOURCE: OUTBOUND SMS MESSAGE REQUESTS ..................................................................................... 42
5.10.1 Request URI variables .................................................................................................................................... 42
5.10.2 Response Codes ............................................................................................................................................ 43
5.10.2.1 HTTP Response Codes ................................................................................................................................ 43
5.10.2.2 Exception fault codes .................................................................................................................................. 43
5.10.3 GET .................................................................................................................................................................. 43
5.10.3.1 Example (Informative) ................................................................................................................................ 43
5.10.3.1.1 Request .................................................................................................................................................... 43
5.10.3.1.2 Response ................................................................................................................................................ 43
5.10.4 PUT ................................................................................................................................................................. 44
5.10.5 POST ............................................................................................................................................................... 44
5.10.5.1 Example 1: returning representation of created resource in response (Informative) ................................. 44
5.10.5.1.1 Request .................................................................................................................................................... 44
5.10.5.1.2 Response ................................................................................................................................................ 45
5.10.5.2 Example 2: returning location of created resource in response (Informative) ............................................ 45
5.10.5.2.1 Request .................................................................................................................................................... 45
5.10.5.2.2 Response ................................................................................................................................................ 46
5.10.5.3 Example 3: serviceException in case of single address or all multiple addresses failure (Informative) .... 46
5.10.5.3.1 Request .................................................................................................................................................... 46
5.10.5.3.2 Response ................................................................................................................................................ 47
5.10.5.4 Example 4: multiple addresses partial success, with deliveryInfoList in response (Informative) ............... 47
5.10.5.4.1 Request .................................................................................................................................................... 47
5.10.5.4.2 Response ................................................................................................................................................ 47
5.10.5.5 Example 5: multiple addresses partial success, without deliveryInfoList in response (Informative) ......... 48
5.10.5.5.1 Request .................................................................................................................................................... 48
5.10.5.5.2 Response ................................................................................................................................................ 48
5.10.6 DELETE ........................................................................................................................................................... 49
5.11 RESOURCE: OUTBOUND SMS MESSAGE REQUEST AND DELIVERY STATUS ............................................... 49
5.11.1 Request URI variables .................................................................................................................................... 49
5.11.2 Response Codes ............................................................................................................................................ 50
5.11.2.1 HTTP Response Codes ................................................................................................................................ 50
5.11.2.2 Exception fault codes .................................................................................................................................. 50
5.11.3 GET .................................................................................................................................................................. 50
5.11.3.1 Example (Informative) ................................................................................................................................ 50
5.11.3.1.1 Request .................................................................................................................................................... 50
5.11.3.1.2 Response ................................................................................................................................................ 50
5.11.4 PUT ................................................................................................................................................................. 51
5.11.5 POST ............................................................................................................................................................... 51
5.11.6 DELETE ........................................................................................................................................................... 51
5.12 RESOURCE: OUTBOUND SMS MESSAGE DELIVERY STATUS .......................................................................... 51
5.12.1 Request URI variables .................................................................................................................................... 51
5.12.2 Response Codes ............................................................................................................................................ 51
5.12.2.1 HTTP Response Codes ................................................................................................................................ 51
5.12.2.2 Exception fault codes .................................................................................................................................. 51
5.12.3 GET .................................................................................................................................................................. 52
5.12.3.1 Example (Informative) ................................................................................................................................ 52
5.12.3.1.1 Request .................................................................................................................................................... 52
5.12.3.1.2 Response ................................................................................................................................................ 52
5.12.4 PUT ................................................................................................................................................................. 52
5.13 RESOURCE: OUTBOUND SMS MESSAGE DELIVERY NOTIFICATION SUBSCRIPTIONS

5.13.1 Request URI variables
5.13.2 Response Codes
5.13.2.1 HTTP Response Codes
5.13.2.2 Exception fault codes
5.13.3 GET
5.13.3.1 Example (Informative)
5.13.3.1.1 Request
5.13.3.1.2 Response
5.13.4 PUT
5.13.5 POST
5.13.5.1 Example (Informative)
5.13.5.1.1 Request
5.13.5.1.2 Response
5.13.6 DELETE

5.14 RESOURCE: INDIVIDUAL OUTBOUND SMS MESSAGE DELIVERY NOTIFICATION SUBSCRIPTION

5.14.1 Request URI variables
5.14.2 HTTP Response Codes
5.14.2.1 HTTP Response Codes
5.14.2.2 Exception fault codes
5.14.3 GET
5.14.3.1 Example (Informative)
5.14.3.1.1 Request
5.14.3.1.2 Response
5.14.4 PUT
5.14.5 POST
5.14.6 DELETE
5.14.6.1 Example (Informative)
5.14.6.1.1 Request
5.14.6.1.2 Response

5.15 RESOURCE: CLIENT NOTIFICATION ABOUT OUTBOUND SMS MESSAGE DELIVERY STATUS

5.15.1 Request URI variables
5.15.2 HTTP Response Codes
5.15.3 GET
5.15.4 PUT
5.15.5 POST
5.15.5.1 Example (Informative)
5.15.5.1.1 Request
5.15.5.1.2 Response
5.15.6 DELETE

APPENDIX A. CHANGE HISTORY (INFORMATIVE)

A.1 APPROVED VERSION HISTORY

A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)

B.1 SCR FOR PARLAYREST.SMS SERVER

B.1.1 SCR for ParlayREST.SMS.Inbound.Registration Server
B.1.2 SCR for ParlayREST.SMS.Inbound.Registration.RetrieveDelete Server
B.1.3 SCR for ParlayREST.SMS.Inbound.Sms Server
B.1.4 SCR for ParlayREST.SMS.Inbound.Subscr Server
B.1.5 SCR for ParlayREST.SMS.Inbound.Individual.Subscr Server
B.1.6 SCR for ParlayREST.SMS.Inbound.Notifications Server
B.1.7 SCR for ParlayREST.SMS.Outbound Server
B.1.8 SCR for ParlayREST.SMS.Outbound.MsgAndDeliveryStatus Server
B.1.9 SCR for ParlayREST.SMS.Outbound.DeliveryStatus Server
B.1.10 SCR for ParlayREST.SMS.Outbound.Subscriptions Server
B.1.11 SCR for ParlayREST.SMS.Inbound.Outbound.Subscr Server
B.1.12 SCR for ParlayREST.SMS.Outbound.DeliveryStatus.Notifications Server

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

C.1 SEND A SMS TO A TERMINAL
C.1.1 Example (Informative)
C.1.1.1 Request
C.1.1.2 Response
C.2 START DELIVERY RECEIPT NOTIFICATION
C.2.1 Example (Informative)
C.2.1.1 Request
C.2.1.2 Response
C.3 START SMS NOTIFICATION
C.3.1 Example (Informative)
C.3.1.1 Request
C.3.1.2 Response

APPENDIX D. JSON EXAMPLES (INFORMATIVE)

D.1 INBOUND MESSAGE DELIVERY (SECTION 5.4.3.1)
D.2 MAXBATCHSIZE EXCEEDING THE ALLOWED SIZE (SECTION 5.4.3.2)
D.3 RETRIEVE AND DELETE USING REGISTRATION (SECTION 5.5.5.1)
D.4 INBOUND MESSAGES FOR A GIVEN REGISTRATION (SECTION 5.6.3.1)
D.5 INVALID (NON-EXISTING) MESSAGEID (SECTION 5.6.3.2)
D.6 REMOVE MESSAGE FROM GATEWAY STORAGE (SECTION 5.6.6.1)
D.7 READ ACTIVE SUBSCRIPTIONS (SECTION 5.7.3.1)
D.8 CREATE NEW MESSAGE SUBSCRIPTION (SECTION 5.7.5.1)
D.9 RETURNING THE LOCATION OF CREATED RESOURCE (SECTION 5.7.5.2)
D.10 READ INDIVIDUAL SUBSCRIPTION (SECTION 5.8.3.1)
D.11 DELETE A SUBSCRIPTION (SECTION 5.8.6.1)
D.12 NOTIFY CLIENT ABOUT MESSAGE ARRIVAL (SECTION 5.9.5.1)
D.13 RETRIEVE LIST OF PENDING OUTBOUND MESSAGES (SECTION 5.10.3.1)
D.14 CREATE OUTBOUND MESSAGE, RETURNING A REPRESENTATION OF CREATED RESOURCE (SECTION 5.10.5.1)
D.15 CREATE OUTBOUND MESSAGE, RETURNING THE LOCATION OF CREATED RESOURCE (SECTION 5.10.5.2)
D.16 SERVICEEXCEPTION IN CASE OF SINGLE ADDRESS OR ALL MULTIPLE ADDRESSES FAILURE (SECTION 5.10.5.3)
D.17 MULTIPLE ADDRESSES PARTIAL SUCCESS, WITH DELIVERYINFOList IN RESPONSE (SECTION 5.10.5.4)
D.18 MULTIPLE ADDRESSES PARTIAL SUCCESS, WITHOUT DELIVERYINFOList IN RESPONSE (SECTION 5.10.5.5)
D.19 GET MESSAGE DELIVERY STATUS (SECTION 5.11.3.1)
D.20 GET MESSAGE DELIVERY STATUS (SECTION 5.12.3.1)
D.21 READ DELIVERY NOTIFICATION SUBSCRIPTIONS (SECTION 5.13.3.1)
D.22 CREATE DELIVERY NOTIFICATION SUBSCRIPTION (SECTION 5.13.5.1)
D.23 READ DELIVERY NOTIFICATION SUBSCRIPTION (SECTION 5.14.3.1)
D.24 DELETE SUBSCRIPTION FOR A CLIENT (SECTION 5.14.6.1)
D.25 NOTIFY CLIENT ABOUT MESSAGE DELIVERY STATUS (SECTION 5.15.5.1)

APPENDIX E. PARLAY X OPERATIONS MAPPING (INFORMATIVE)

Figures

Figure 1: Resource structure defined by this specification
Figure 2: Send SMS and check the delivery status
Figure 3: Inbound SMS message delivery (push mode)
Figure 4: Inbound SMS message delivery (polling mode)
1. Scope

This specification defines an HTTP protocol binding for an abstract API using the REST architectural style, based on an existing OMA enabler namely the Short Messaging Service, as defined in [3GPP 29.199-4].
2. References

2.1 Normative References


[W3C-URLENC] W3C HTML 2.0 Specification, form-urlencoded Media Type, URL: http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1


2.2 Informative References


3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>EMS</td>
<td>Enhanced Message Service</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>MSISDN</td>
<td>Mobile Subscriber ISDN Number</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>REST</td>
<td>REpresentational State Transfer</td>
</tr>
<tr>
<td>SCR</td>
<td>Static Conformance Requirements</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>TS</td>
<td>Technical Specification</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition</td>
</tr>
</tbody>
</table>
4. Introduction

The ParlayREST Technical Specification for SMS contains the HTTP protocol binding for the Parlay X Short 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 Short Messaging Service ParlayREST API specification supports the following operations:

- Send text 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 notifications for outbound messages based on given criteria (online)
- Delete subscriptions for notifications for outbound messages (online)
- Retrieve message content
- Confirm message retrieval by deleting message (execute delete command)
5. Short Messaging Service (SMS) API definition

This section is organized to support a comprehensive understanding of the SMS 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 of the API 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 SMSs (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). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

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

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

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

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

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

5.1 Resources Summary

This section summarizes all the resources used by the SMS 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.
Figure 1: Resource structure defined by this specification

The following table gives a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.
### Purpose: Inbound messages for periodic polling (based on a provisioning step configuration)

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Base URL:</strong> http://{serverRoot}/{apiVersion}/smsmessaging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inbound SMS messages requests for a given registration</td>
<td>/inbound/registrations/{registrationId}/messages</td>
<td>InboundSMSMessageList</td>
<td>GET: read one or more messages from gateway storage</td>
</tr>
<tr>
<td></td>
<td>Note: Used by clients that periodically poll for incoming messages. Retrieval criteria have to be provisioned in advance.</td>
<td></td>
<td>POST: no, PUT: no, DELETE: no</td>
</tr>
<tr>
<td></td>
<td>/inbound/registrations/{registrationId}/retrieveAndDeleteMessages</td>
<td>InboundSMSMessageList (used for POST response)</td>
<td>GET: no, POST: pops one or more messages from the gateway storage (remove it if successful), PUT: no, DELETE: no</td>
</tr>
<tr>
<td></td>
<td>/inbound/registrations/{registrationId}/messages/{messageId}</td>
<td>InboundSMSMessage (used for POST response)</td>
<td>GET: read one message from gateway storage, PUT: delete one message from gateway storage Note: Messages are automatically deleted after a certain time.</td>
</tr>
</tbody>
</table>

### Purpose: Subscriptions Management for Inbound Messages

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Base URL:</strong> http://{serverRoot}/{apiVersion}/smsmessaging</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>URL</td>
<td>Data Structures</td>
<td>HTTP verbs</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Base URL:</td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td>http://{serverRoot}/{apiVersion}/smsmessaging</td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SubscriptionList (used for GET)</td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subscription (used for POST)</td>
<td>DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>common:ResourceReference (optional alternative for POST response)</td>
<td></td>
</tr>
<tr>
<td>Inbound SMS message subscriptions</td>
<td>/inbound/subscriptions</td>
<td>Subscription</td>
<td>read all active subscriptions</td>
</tr>
<tr>
<td>Individual inbound SMS message subscription</td>
<td>/inbound/subscriptions/{subscriptionId}</td>
<td>Subscription</td>
<td>read individual subscription</td>
</tr>
</tbody>
</table>

**Purpose: Callback notifications for Inbound messages**

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client notification about inbound SMS message</td>
<td>&lt;specified by the client&gt; when subscription is created or during provisioning process&gt;</td>
<td>InboundSMSMessageNotification</td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>notifies client about new inbound message</td>
</tr>
</tbody>
</table>

**Purpose: Sending SMS and obtaining the Delivery Status**

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Base URL:</td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td>http://{serverRoot}/{apiVersion}/smsmessaging</td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outbound SMS message requests</td>
<td>/outbound/{senderAddress}/requests</td>
<td>OutboundSMSMessageRequestList (used for GET)</td>
<td>read all pending outbound message requests</td>
</tr>
<tr>
<td>Outbound SMS message request and delivery status</td>
<td>/outbound/{senderAddress}/requests/{requestId}</td>
<td>OutboundSMSMessageRequest</td>
<td>read a certain sent SMS message, including the delivery status</td>
</tr>
<tr>
<td>Outbound SMS message delivery status</td>
<td>/outbound/{senderAddress}/requests/{requestId}/deliveryInfos</td>
<td>DeliveryInfoList</td>
<td>read delivery status for the individual outbound message</td>
</tr>
</tbody>
</table>
### Purpose: Subscription Management for Outbound Message Delivery Status

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound SMS message delivery notification subscriptions</td>
<td>/outbound/{senderAddress}/subscriptions</td>
<td>DeliveryReceiptSubscriptionList (used for GET) DeliveryReceiptSubscription (used for POST) common:ResourceReference (optional alternative for POST response)</td>
<td>read all outbound SMS subscriptions create new delivery receipt subscription no no</td>
</tr>
<tr>
<td>Individual outbound SMS message delivery notification subscription</td>
<td>/outbound/{senderAddress}/subscriptions/{subscriptionId}</td>
<td>DeliveryReceiptSubscription</td>
<td>read an individual outbound SMS subscription no no remove subscription and stop corresponding delivery receipt notifications</td>
</tr>
</tbody>
</table>

### Purpose: Callback notifications for Outbound Message Delivery Status

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client notification about outbound SMS message delivery status</td>
<td>&lt;specified by the client&gt; when outbound request is submitted&gt;</td>
<td>DeliveryInfoNotification</td>
<td>no Notifies client about delivery status of outgoing requests no no</td>
</tr>
</tbody>
</table>

### 5.2 SMS ParlayREST API Data Structures

The namespace for the ShortMessaging data types is:

```
urn:oma:xml:rest:sms: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: InboundSMSMessageList

List of received SMS.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inboundSMSMessage</td>
<td>InboundSMSMessage [0..unbounded]</td>
<td>Yes</td>
<td>It may contain an array of messages received according to the specified {registrationId}.</td>
</tr>
<tr>
<td>totalNumberOfPendingMessages</td>
<td>xsd:int</td>
<td>Yes</td>
<td>Total number of messages in the gateway storage waiting for retrieval at the time of the request</td>
</tr>
<tr>
<td>numberOfMessagesInThisBatch</td>
<td>xsd:int</td>
<td>Yes</td>
<td>Number of the messages included in the response (part of the totalNumberOfPendingMessages)</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Self referring URL</td>
</tr>
</tbody>
</table>

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

5.2.2  Type: InboundSMSMessage

Individual incoming SMS.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destinationAddress</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Number associated with the invoked Message service, i.e. the destination address used by the terminal to send the message.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Indicates message senderAddress.</td>
</tr>
<tr>
<td>message</td>
<td>xsd:string</td>
<td>No</td>
<td>Text of the message</td>
</tr>
<tr>
<td>dateTime</td>
<td>xsd:dateTime</td>
<td>Yes</td>
<td>Time when message was received by operator</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Links to other resources that are in relationship with the resource</td>
</tr>
<tr>
<td>messageId</td>
<td>xsd:string</td>
<td>Yes</td>
<td>OPTIONAL server-generated message Identifier</td>
</tr>
</tbody>
</table>

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

5.2.3  Type: InboundSMSMessageNotification
callbackData    xsd:string    Yes    CallbackData as passed by the application during the associated Send SMS operation. See [REST_TS_Common], section 6.2.4.

inboundSMSMessage    InboundSMSMessage    No    SMS message

link    common:Link[0..unbounded]    Yes    Link to other resources. For example we can have a link to the original outbound message request.

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

5.2.4 Type: SubscriptionList

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscription</td>
<td>Subscription[0..unbounded]</td>
<td>Yes</td>
<td>It may contain an array of Subscription</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
</tbody>
</table>

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

5.2.5 Type: Subscription

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callbackReference</td>
<td>common:CallbackReference</td>
<td>No</td>
<td>Client's Notification URL and OPTIONAL callbackData</td>
</tr>
<tr>
<td>destinationAddress</td>
<td>xsd:anyURI [1..unbounded]</td>
<td>No</td>
<td>the destination address of the short message</td>
</tr>
<tr>
<td>criteria</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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</td>
</tr>
</tbody>
</table>
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_TSEnterprise], 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_TSEnterprise] provides a recommendation regarding the generation of the value of this field.

### 5.2.6 Type: InboundSMSMessageRetrieveAndDeleteRequest

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrievalOrder</td>
<td>RetrievalOrder</td>
<td>Yes</td>
<td>Specifies order in which messages should be retrieved if there are more then one pending</td>
</tr>
<tr>
<td>maxBatchSize</td>
<td>xsd:int</td>
<td>Yes</td>
<td>Specifies maximum number of messages to be returned in the response</td>
</tr>
</tbody>
</table>

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

### 5.2.7 Type: OutboundSMSMessageRequestList

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outboundSMSMessageRequest</td>
<td>OutboundSMSMessageRequest [0..unbounded]</td>
<td>Yes</td>
<td>The messages that have been sent by the API Consumer and still exist in the server (and hence the messages that can be accessed by the API Consumer). Messages exist in the server for a little time after reaching their final Delivery Status</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
</tbody>
</table>

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document
A root element named outboundSMSMessageRequestList of type OutboundSMSMessageRequestList is allowed in request and/or response bodies.

### 5.2.8 Type: OutboundSMSMessageRequest

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>xsd:anyURI [1..unbounded]</td>
<td>No</td>
<td>Destination addresses for the Message.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The address of the sender to whom a responding SMS may be sent. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '%2B' followed by the country code before the subscriber number; e.g. tel:%2B447990123456.</td>
</tr>
<tr>
<td>senderName</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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).</td>
</tr>
<tr>
<td>charging</td>
<td>common:Charging Information</td>
<td>Yes</td>
<td>Charging to apply to this message.</td>
</tr>
<tr>
<td>receiptRequest</td>
<td>common:CallbackReference</td>
<td>Yes</td>
<td>It defines the application endpoint and OPTIONAL callbackData that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible.</td>
</tr>
<tr>
<td>outboundSMSTextMessage</td>
<td>OutboundSMSTextMessage</td>
<td>Choice</td>
<td>Included if a SMSText is being Sent.</td>
</tr>
<tr>
<td>outboundSMSBinaryMessage</td>
<td>OutboundSMSBinaryMessage</td>
<td>Choice</td>
<td>Included if a SMSBinary is being Sent.</td>
</tr>
<tr>
<td>outboundSMSLogoMessage</td>
<td>OutboundSMSLogoMessage</td>
<td>Choice</td>
<td>Included if a SMSLogo is being Sent.</td>
</tr>
<tr>
<td>outboundSMSRingToneMessage</td>
<td>OutboundSMSRingToneMessage</td>
<td>Choice</td>
<td>Included if a SMSRingtone is being Sent.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to</td>
</tr>
</tbody>
</table>
recover from communication failures during resource creation and therefore avoids re-sending the message in such situations.

In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.

| resourceURL | xsd:anyURI | Yes | Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests. |
| link | common:Link[0..unbounded] | Yes | Links to other resources that are in relationship with the resource |
| deliveryInfoList | DeliveryInfoList | Yes | The Delivery Information (filled in by the server) |

XSD modelling use a “choice” to select either a SMSText, a SMSBinary, a SMSLogo or a SMSRingTone.

Note: SMSBinary is supported in order to facilitate legacy applications that may send SMS in binary format (e.g. using SMPP). Underlying implementations need to be aware whether SMSCs and/or final destination mobile phones can handle such messages without unforeseen side effects. Implementations MUST support Service Provider policies to accept or reject the handling of a binarySMS message (POL0001: Policy error SHALL be used in case the message is rejected, see [REST_TS_Common]).

A root element named outboundSMSMessageRequest of type OutboundSMSMessageRequest 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: OutboundSMSTextMessage

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>xsd:string</td>
<td>No</td>
<td>Short message content.</td>
</tr>
</tbody>
</table>

### 5.2.10 Type: OutboundSMSBinaryMessage

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>message</td>
<td>xsd:base64Binary</td>
<td>No</td>
<td>Short message content in binary format.</td>
</tr>
</tbody>
</table>
### 5.2.11 Type: OutboundSMSLogoMessage

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image</td>
<td>xsd:base64Binary</td>
<td>No</td>
<td>The image in jpeg, gif or png format. The image will be scaled to the proper format</td>
</tr>
<tr>
<td>smsFormat</td>
<td>SMSFormat</td>
<td>No</td>
<td>Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'</td>
</tr>
</tbody>
</table>

### 5.2.12 Type: OutboundSMSRingToneMessage

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ringTone</td>
<td>xsd:string</td>
<td>No</td>
<td>The ring tone in RTX format. Note: In the RTX Ring tone 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.</td>
</tr>
<tr>
<td>smsFormat</td>
<td>SmsFormat</td>
<td>No</td>
<td>Conversion to be applied to the message prior to delivery. Possible values are: 'Ems' or 'SmartMessaging'</td>
</tr>
</tbody>
</table>

### 5.2.13 Type: DeliveryInfoList

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Self referring URL</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Links to other resources that are in relationship with the resource.</td>
</tr>
<tr>
<td>deliveryInfo</td>
<td>DeliveryInfo[1..unbounded]</td>
<td>No</td>
<td>Delivery Information</td>
</tr>
</tbody>
</table>

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

### 5.2.14 Type: DeliveryInfoNotification

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callbackData</td>
<td>xsd:string</td>
<td>Yes</td>
<td>CallbackData if passed by the application in the receiptRequest element during the associated Send SMS operation. See [REST_TS_Common], section 6.2.4.</td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
<td>Optional</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>deliveryInfo</td>
<td>DeliveryInfo[1…unbounded]</td>
<td>No</td>
<td>Delivery Information</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Links to other resources that are in relationship to the current resource.</td>
</tr>
</tbody>
</table>

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

### 5.2.15 Type: DeliveryInfo

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Outbound message destination address</td>
</tr>
<tr>
<td>deliveryStatus</td>
<td>DeliveryStatus</td>
<td>No</td>
<td>Indicates the delivery result for the destination address.</td>
</tr>
<tr>
<td>description</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Used together with Delivery Status (e.g. DeliveryImpossible) to provide additional information.</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Links to other resources that are in relationship with the resource. For example we can have a link to the original outbound message request.</td>
</tr>
</tbody>
</table>

### 5.2.16 Type: DeliveryReceiptSubscriptionList

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Self referring URL</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Link to other resources that are in relationship with the resource</td>
</tr>
<tr>
<td>deliveryReceiptSubscription</td>
<td>DeliveryReceiptSubscription[0…unbounded]</td>
<td>Yes</td>
<td>Delivery Subscription Information</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callbackReference</td>
<td>common:CallbackReference</td>
<td>No</td>
<td>Notification endpoint definition</td>
</tr>
<tr>
<td>filterCriteria</td>
<td>xsd:string</td>
<td>No</td>
<td>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.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>resourceURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>Self referring URL. SHALL NOT be included in POST requests, MUST be included in responses to any HTTP method that returns an entity body, and in PUT requests.</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Link to other resources that are in relationship with the resource.</td>
</tr>
</tbody>
</table>

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.

5.2.18 Enumeration: DeliveryStatus

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeliveredToTerminal</td>
<td>Successful delivery to Terminal.</td>
</tr>
<tr>
<td>DeliveryUncertain</td>
<td>Delivery status unknown: e.g. because it was handed off to another network.</td>
</tr>
<tr>
<td>DeliveryImpossible</td>
<td>Unsuccessful delivery; the message could not be delivered before it expired.</td>
</tr>
<tr>
<td>MessageWaiting</td>
<td>The message is still queued for delivery. This is a temporary state, pending transition to one of the preceding states.</td>
</tr>
</tbody>
</table>
DeliveredToNetwork | Successful delivery to the network enabler responsible for distributing the short 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 SendMessageRequest is not supported.

### 5.2.19 Enumeration: SmsFormat

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ems</td>
<td>EMS conversion</td>
</tr>
<tr>
<td>SmartMessaging</td>
<td>SmartMessaging conversion</td>
</tr>
</tbody>
</table>

### 5.2.20 Enumeration: RetrievalOrder

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OldestFirst</td>
<td>Retrieve in the order from oldest to newest</td>
</tr>
<tr>
<td>NewestFirst</td>
<td>Retrieve in the order from newest to oldest</td>
</tr>
</tbody>
</table>

### 5.2.21 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):

- InboundSMSMessage
- InboundSMSMessageList
- Subscription
- SubscriptionList
- OutboundSMSMessageRequest
- OutboundSMSMessageRequestList
- DeliveryInfoList
- DeliveryReceiptSubscription
- DeliveryReceiptSubscriptionList

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

5.3.1 Send SMS and check the delivery status

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

The resources:

- To send a short message, create new resource under
  \[
  \text{http://[serverRoot]/[apiVersion]/smsmessaging/outbound/[senderAddress]/requests}
  \]

- 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
    \[
    \text{http://[serverRoot]/[apiVersion]/smsmessaging/outbound/[senderAddress]/requests/[requestId]}
    \]
  b. directly read the resource
    \[
    \text{http://[serverRoot]/[apiVersion]/smsmessaging/outbound/[senderAddress]/requests/[requestId]/deliveryInfos}
    \]

![Sequence Diagram](image)

**Figure 2: Send SMS and check the delivery status**

Outline of the flows:

1. An application initiates the creation of new outbound SMS 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 SMS message delivery (push mode)

This figure below shows a scenario for starting notification of inbound SMS 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 SMS messages, create new resource under
  \[http://\{serverRoot\} /\{apiVersion\}/smsmessaging/inbound/subscriptions\]
- To notify the application about the message arrival, POST a notification to the client supplied notifyURL
- To stop the subscription to notifications, delete the resource
  \[http://\{serverRoot\} /\{apiVersion\}/smsmessaging/inbound/subscriptions/\{subscriptionId\}\]

![Diagram](image)

**Figure 3: Inbound SMS message delivery (push mode)**

Outline of the flows:

1. An application subscribes to notifications for inbound messages using POST and receives the resulting resourceURL containing the subscriptionId.
2. When the message which satisfies the specified criteria arrives, the REST service notifies the application of the incoming message using POST so that the application receives the message.
3. The application stops the notification subscription using DELETE with a resource URL containing the subscriptionId.

### 5.3.3 Inbound SMS message delivery (polling mode)

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

The resources:

- To retrieve incoming messages satisfying the criteria set up in advance, get the resource
  
  \[
  \text{http://[serverRoot]/[apiVersion]/smsmessaging/inbound/registrations/}[\text{[registrationId]}]/messages
  \]

- To remove one message from the storage, delete the resource
  
  \[
  \text{http://[serverRoot]/[apiVersion]/smsmessaging/inbound/registrations/}[\text{[registrationId]}]/messages/\text{[messageId]}
  \]

![Diagram of Inbound SMS message delivery (polling mode)](image)

Figure 4: Inbound SMS message delivery (polling mode)

Outline of the flows:

1. In advance, the notification of SMS 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 and receives the messages.
3. The application removes one of the messages from gateway storage using DELETE with a resource URL containing the messageId.
5.4 Resource: Inbound SMS message requests for a given registration

The resource used is:

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

This resource is used for checking for incoming messages using a retrieval criterion that is setup in advance for a particular client (offline - during provisioning process: sms short codes, etc).

5.4.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>registrationId</td>
<td>reference to the off-line retrieval criteria provisioned in advance and known to the client application. Analogous to ParlayX registrationIdentifier</td>
</tr>
</tbody>
</table>

5.4.2 Response Codes

5.4.2.1 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 Short Messaging, see [3GPP 29.199-4].

5.4.3 GET

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

Request URL parameters are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxBatchSize</td>
<td>xsd:int</td>
<td>Yes</td>
<td>Specifies maximum number of messages to be returned in the response</td>
</tr>
<tr>
<td>retrievalOrder</td>
<td>RetrievalOrder</td>
<td>Yes</td>
<td>Specifies order in which messages should be retrieved is there are more then one pending</td>
</tr>
</tbody>
</table>

5.4.3.1 Example 1: regular request (Informative)

5.4.3.1.1 Request

GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages?maxBatchSize=2 HTTP/1.1
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"?>
<sms:inboundSMSMessageList xmlns:sms="urn:oma:xml:rest:sms:1">
  <inboundSMSMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN2</senderAddress>
    <message>First simple message</message>
    <dateTime>2009-11-19T12:00:00</dateTime>
    <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1}</resourceURL>
    <messageId>{messageId1}</messageId>
  </inboundSMSMessage>
  <inboundSMSMessage>
    <destinationAddress>MSISDN1</destinationAddress>
    <senderAddress>MSISDN3</senderAddress>
    <message>Second simple message</message>
    <dateTime>2009-11-19T12:00:00</dateTime>
    <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId2}</resourceURL>
    <messageId>{messageId2}</messageId>
  </inboundSMSMessage>
  <!-- further instances of InboundSMSMessage if needed -->
  <totalNumberOfPendingMessages>20</totalNumberOfPendingMessages>
  <numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages</resourceURL>
</sms:inboundSMSMessageList>

5.4.3.2 Example 2: maxBatchSize exceeding the allowed size (Informative)

5.4.3.2.1 Request

GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages?maxBatchSize=5000 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="InboundSMSMessageList" href="http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages?maxBatchSize=5000" />
  <policyException>
    <messageId>POL0001</messageId>
    <text>A policy error occurred. Error code is maxBatchSize exceeded. The maximum allowed maxBatchSize is %1.</text>
    <variables>20</variables>
  </policyException>
</common:requestError>

5.4.4 PUT

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

5.4.5 POST

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

5.4.6 DELETE

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

5.5 Resource: Inbound SMS messages Retrieve and Delete using registration

The resource used is:

http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/retrieveAndDeleteMessages

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

5.5.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>registrationId</td>
<td>reference to the off-line retrieval criteria provisioned in advance and known to the client application. Analogous to ParlayX registrationIdentifier</td>
</tr>
</tbody>
</table>
5.5.2 Response Codes

5.5.2.1 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 Short Messaging, see [3GPP 29.199-4].

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 a 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/registrations resource.

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

3) Messages would be automatically deleted from gateway storage following a successful POST, after a maximum time interval as defined by a service policy.

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

5.5.5.1 Example (Informative)

5.5.5.1.1 Request

POST .../(apiVersion)/smsmessaging/inbound/registrations/(registrationId)/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"?>
<sms:inboundSMSMessageRetrieveAndDeleteRequest xmlns:sms="urn:oma:xml:rest:sms:1">
<retrievalOrder>OldestFirst</retrievalOrder>
<maxBatchSize>3</maxBatchSize>
</sms:inboundSMSMessageRetrieveAndDeleteRequest>
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"?>
<sms:inboundSMSMessageList xmlns:sms="urn:oma:xml:rest:sms:1">
  <!-- SMS -->
  <inboundSMSMessage>
    <destinationAddress>{shortCode1}</destinationAddress>
    <senderAddress>{MSISDN1}</senderAddress>
    <message>First simple message</message>
    <messageId>{messageId1}</messageId>
    <!-- no message resourceURL because SMS will be deleted from server immediately after operation is completed -->
  </inboundSMSMessage>
  <!-- SMS -->
  <inboundSMSMessage>
    <destinationAddress>{shortCode2}</destinationAddress>
    <senderAddress>{MSISDN2}</senderAddress>
    <message>Second simple message</message>
    <messageId>{messageId2}</messageId>
    <!-- no message resourceURL because SMS will be deleted from server immediately after operation is completed -->
  </inboundSMSMessage>
  <totalNumberOfPendingMessages>200</totalNumberOfPendingMessages>
  <numberOfMessagesInThisBatch>2</numberOfMessagesInThisBatch>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}</resourceURL>
</sms:inboundSMSMessageList>

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: Inbound SMS message for a given registration

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

This resource provides access to individual inbound SMS message stored by 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.

5.6.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>registrationId</td>
<td>reference to the off-line retrieval criteria provisioned in advance and known to the</td>
</tr>
</tbody>
</table>
5.6.2 Response Codes

5.6.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.6.3 GET

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.

5.6.3.1 Example 1: regular request (Informative)

5.6.3.1.1 Request

GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId}?resFormat=XML

HTTP/1.1
Host: example.com:80

The sample request above also illustrates the use of the “resFormat” parameter.

5.6.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:inboundSMSMessage xmlns:sms="urn:oma:xml:rest:sms:1">
  <destinationAddress>MSISDN1</destinationAddress>
  <senderAddress>MSISDN2</senderAddress>
  <message>First simple message</message>
  <dateTime>2009-11-19T12:00:00</dateTime>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId}</resourceURL>
  <messageId>{messageId}</messageId>
</sms:inboundSMSMessage>

5.6.3.2 Example 2: invalid (non-existing) messageId (Informative)

5.6.3.2.1 Request

GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1} HTTP/1.1
Accept: application/xml
Host: example.com:80
5.6.3.2.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <link rel="InboundSMSMessage" href="http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1}" />
  <serviceException>
    <messageId>SVC0002</messageId>
    <text>Invalid input value. The requested messageId %1 does not exist.</text>
    <variables>{messageId1}</variables>
  </serviceException>
</common:requestError>

5.6.4 PUT

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

5.6.5 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].

5.6.6 DELETE

Confirms message delivery and removes the message from the storage on the gateway.

5.6.6.1 Example (Informative)

5.6.6.1.1 Request

DELETE .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1} HTTP/1.1
Accept: application/xml
Host: example.com:80

5.6.6.1.2 Response

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

5.7 Resource: Inbound SMS message subscriptions

The resource used is: http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions
This resource gives access to inbound subscriptions for a particular client.

### 5.7.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)</td>
</tr>
</tbody>
</table>

### 5.7.2 Response Codes

#### 5.7.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

#### 5.7.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4].

### 5.7.3 GET

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

#### 5.7.3.1 Example (Informative)

**5.7.3.1.1 Request**

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

**5.7.3.1.2 Response**

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sms:subscriptionList xmlns:sms="urn:oma:xml:rest:sms:1">
  <subscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <destinationAddress>680180999</destinationAddress>
    <criteria>Urgent*</criteria>
    <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId1}</resourceURL>
  </subscription>
</sms:subscriptionList>
```
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, POST’ field in the response as per section 14.7 of [RFC 2616].

5.7.5 POST

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

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

5.7.5.1.1 Request

```xml
POST .../(apiVersion)/smsmessaging/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"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
    <callbackReference>
        <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
        <callbackData>12345</callbackData>
    </callbackReference>
    <destinationAddress>+34680180999</destinationAddress>
    <criteria>Urgent*</criteria>
</sms:subscription>
```

5.7.5.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
    <callbackReference>
        <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
        <callbackData>12345</callbackData>
    </callbackReference>
    <destinationAddress>+34680180999</destinationAddress>
    <criteria>Urgent*</criteria>
</sms:subscription>
```
<criteria>Urgent*</criteria>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId1}</resourceURL>
</sms:subscription>

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

5.7.5.2.1 Request

POST .../{apiVersion}/smsmessaging/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"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
</sms:subscription>

5.7.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/smsmessaging/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}/smsmessaging/inbound/subscriptions/{subscriptionId1}</resourceURL>
</common:resourceReference>

5.7.6 DELETE

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

5.8 Resource: Individual inbound SMS message subscription

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

This resource controls individual subscription for inbound messages and gives access to individual subscription for a particular client.
5.8.1 Request URI variables

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

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

5.8.2 Response Codes

5.8.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.8.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4]

5.8.3 GET

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

5.8.3.1 Example (Informative)

5.8.3.1.1 Request

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

5.8.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Urgent*</criteria>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscription/{subscriptionId}</resourceURL>
</sms:subscription>
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 delete a subscription for the particular client.

5.8.6.1 Example (Informative)

5.8.6.1.1 Request

DELETE .../(apiVersion)/smsmessaging/inbound/subscriptions/{subscriptionId} 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: Client notification about inbound SMS message

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

5.9.1 Request URI variables

Client provided.

5.9.2 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.9.3 GET

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

5.9.4 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].
5.9.5 POST

This operation is used to notify client about message arrival.

5.9.5.1 Example

5.9.5.1.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:inboundSMSMessageNotification xmlns:sms="urn:oma:xml:rest:sms:1">
    <callbackData>12345</callbackData>
    <inboundSMSMessage>
        <destinationAddress>MSISDN</destinationAddress>
        <senderAddress>sender name</senderAddress>
        <message>First simple message</message>
        <dateTime>2009-11-19T12:00:00</dateTime>
        <messageId>{messageId1}</messageId>
    </inboundSMSMessage>
</sms:inboundSMSMessageNotification>
```

5.9.5.1.2 Response

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

5.9.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405

5.10 Resource: Outbound SMS message requests

The resource used is: `http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests`

This resource is used for sending outbound messages.

5.10.1 Request URI variables

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

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

5.10.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.10.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4].

5.10.3 GET

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

5.10.3.1 Example (Informative)

5.10.3.1.1 Request

GET .../apiVersion/smsmessaging/outbound/{senderAddress}/requests 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"?>
<sms:outboundSMSMessageRequestList xmlns:sms="urn:oma:xml:rest:sms:1">
  <outboundSMSMessageRequest>
    <address>tel:+1350000001</address>
    <senderAddress>tel:+1350000512</senderAddress>
    <outboundSMSTextMessage>
      <message>Let's have a REST.</message>
    </outboundSMSTextMessage>
    <clientCorrelator>67891</clientCorrelator>
    <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId1}</resourceURL>
    <deliveryInfoList>
      <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId1}/deliveryInfos</resourceURL>
      <deliveryInfo>
        <address>tel:+1350000001</address>
        <deliveryStatus>DeliveredToNetwork</deliveryStatus>
      </deliveryInfo>
    </deliveryInfoList>
  </outboundSMSMessageRequest>
  <outboundSMSMessageRequest>
    <address>tel:+1350000991</address>
    <address>tel:+1350000992</address>
  </outboundSMSMessageRequest>
</sms:outboundSMSMessageRequestList>
5.10.4 PUT

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

5.10.5 POST

This operation is used to create outgoing message request.

5.10.5.1 Example 1: returning representation of created resource in response

(Informative)

5.10.5.1.1 Request

POST .../(apiVersion)/smsmessaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
  <senderAddress>tel:+1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest> <-- this is optional -->
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Example Text Message</message>
  </outboundSMSTextMessage>
  <clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequestRequestList>
5.10.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
  <senderAddress>tel:+1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest> <!-- this is optional -->
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Example Text Message</message>
  </outboundSMSTextMessage>
  <clientCorrelator>67893</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
  <deliveryInfoList>
    <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/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>
</sms:outboundSMSMessageRequest>

5.10.5.2 Example 2: returning location of created resource in response (Informative)

5.10.5.2.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
5.10.5.2.2  Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: {serverRoot}/(apiVersion)/smsmessaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: 12345
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)/smsmessaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
</common:resourceReference>

5.10.5.3  Example 3: serviceException in case of single address or all multiple addresses failure  (Informative)

5.10.5.3.1  Request

POST .../(apiVersion)/smsmessaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml; charset=UTF-8
Host: example.com:80

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+135000999</address>
  <senderAddress>tel:+1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest> <!-- this is optional -->
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Example Text Message</message>
  </outboundSMSTextMessage>
  <clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequest>
5.10.5.3.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:common:1">
  <serviceException>
    <messageId>SVC0001</messageId>
    <text>A service error occurred. Error code is %1</text>
    <variables>ERROR-XYZ</variables>
  </serviceException>
</common:requestError>

5.10.5.4 Example 4: multiple addresses partial success, with deliveryInfoList in response
(Informative)

5.10.5.4.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
  <senderAddress>tel:+1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest> <!-- this is optional -->
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Example Text Message</message>
  </outboundSMSTextMessage>
  <clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequest>

5.10.5.4.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
<senderAddress>tel:+1351111999</senderAddress>
<senderName>MyName</senderName>
<receiptRequest> <!-- this is optional -->
<notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
</receiptRequest>
<outboundSMSTextMessage>
<message>Example Text Message</message>
</outboundSMSTextMessage>
<clientCorrelator>67893</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
<deliveryInfoList>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos</resourceURL>
<deliveryInfo>
<address>tel:+1350000001</address>
<deliveryStatus>MessageWaiting</deliveryStatus>
</deliveryInfo>
<deliveryInfo>
<address>tel:+1350000999</address>
<deliveryStatus>DeliveryImpossible</deliveryStatus>
</deliveryInfo>
</deliveryInfoList>
</sms:outboundSMSMessageRequest>

5.10.5.5  Example 5: multiple addresses partial success, without deliveryInfoList in response  (Informative)

5.10.5.5.1  Request

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
<address>tel:+1350000001</address>
<address>tel:+1350000999</address>
<senderAddress>tel:+1351111999</senderAddress>
<senderName>MyName</senderName>
<receiptRequest> <!-- this is optional -->
<notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
</receiptRequest>
<outboundSMSTextMessage>
<message>Example Text Message</message>
</outboundSMSTextMessage>
<clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequest>

5.10.5.5.2  Response

Note: In this case, in order to know the result of sending to individual addresses, the delivery status can be obtained using the GET operation with the requestId, or via notifications (if subscribed).
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: nnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+1350000001</address>
  <address>tel:+1350000999</address>
  <senderAddress>tel:+1351111999</senderAddress>
  <senderName>MyName</senderName>
  <receiptRequest> <!-- this is optional -->
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Example Text Message </message>
  </outboundSMSTextMessage>
  <clientCorrelator>67893</clientCorrelator>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
</sms:outboundSMSMessageRequest>

5.10.6 DELETE

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

5.11 Resource: Outbound SMS message request and delivery status

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

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

5.11.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>senderAddress</td>
<td>identifies client application. Typically SMS SHORT CODE [REST_TS_Common]</td>
</tr>
<tr>
<td>requestId</td>
<td>outbound message request Id generated by server</td>
</tr>
</tbody>
</table>
5.11.2 Response Codes

5.11.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.11.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4].

5.11.3 GET

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

5.11.3.1 Example (Informative)

5.11.3.1.1 Request

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

5.11.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms: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>
</receiptRequest>
<outboundSMSTextMessage>
<message>sent message</message>
</outboundSMSTextMessage>
<clientCorrelator>67893</clientCorrelator>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}</resourceURL>
<deliveryInfoList>
<!-- this is optional -->
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos</resourceURL>
<deliveryInfo>
<address>tel:+1350000001</address>
<deliveryStatus>MessageWaiting</deliveryStatus>
</deliveryInfo>
<deliveryInfo>
  <address>tel:+13500009999</address>
  <deliveryStatus>MessageWaiting</deliveryStatus>
</deliveryInfo>
</deliveryInfoList>
</sms:outboundSMSMessageRequest>

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

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

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

5.12 Resource: Outbound SMS message delivery status
The resource used is:
http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos

This resource is used to request outbound message delivery status.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>senderAddress</td>
<td>identifies client application. Typically SMS SHORT CODE [REST_TS_Common]</td>
</tr>
<tr>
<td>requestId</td>
<td>outbound message request Id generated by server</td>
</tr>
</tbody>
</table>

5.12.2 Response Codes

5.12.2.1 HTTP Response Codes
For HTTP response codes, see [REST_TS_Common].

5.12.2.2 Exception fault codes
For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4].
5.12.3  GET

This operation is used to retrieve outgoing message delivery status.

5.12.3.1  Example  (Informative)

5.12.3.1.1  Request

GET .../(apiVersion)/smsmessaging/outbound/(senderAddress)/requests/{requestId}/deliveryInfos 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"?>
<sms:deliveryInfoList xmlns:sms="urn:oma:xml:rest:sms:1">
  <resourceURL>
    http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos
  </resourceURL>
  <deliveryInfo>
    <address>tel:+1350000001</address>
    <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
  <deliveryInfo>
    <address>tel:+1350000999</address>
    <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
</sms:deliveryInfoList>

5.12.4  PUT

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

5.12.5  POST

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

5.12.6  DELETE

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

5.13  Resource: Outbound SMS message delivery notification subscriptions

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

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

### 5.13.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API clients want to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>senderAddress</td>
<td>identifies client application. Typically SMS SHORT CODE [REST_TS_Common]</td>
</tr>
</tbody>
</table>

### 5.13.2 Response Codes

#### 5.13.2.1 HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

#### 5.13.2.2 Exception fault codes

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see [3GPP 29.199-4].

### 5.13.3 GET

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

#### 5.13.3.1 Example (Informative)

##### 5.13.3.1.1 Request

GET .../{apiVersion}/smsmessaging/outbound/{senderAddress}/subscriptions 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
Date: Thu, 04 Jun 2009 02:51:59 GMT

```xml
```
<callbackData>12345</callbackData>
</callbackReference>
<filterCriteria>0102</filterCriteria>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/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}/smsmessaging/outbound/subscriptions/{subscriptionId}</resourceURL>
</deliveryReceiptSubscription>
</sms:deliveryReceiptSubscriptionList>

5.13.4 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].

5.13.5 POST

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

5.13.5.1 Example (Informative)

5.13.5.1.1 Request

POST .../{apiVersion}/smsmessaging/outbound/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"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:sms:1">
<callbackReference>
<notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
<callbackData>12345</callbackData>
</callbackReference>
<filterCriteria>0102</filterCriteria>
<resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/{subscriptionId}</resourceURL>
</deliveryReceiptSubscription>
</sms:deliveryReceiptSubscription>

Note that this subscription example does not use the clientCorrelator but provides callbackData.

5.13.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location:http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/{subscriptionId}
Date: Thu, 04 Jun 2009 02:51:59 GMT
<?xml version="1.0" encoding="UTF-8"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/{subscriptionId}</resourceURL>
</sms:deliveryReceiptSubscription>

Note that alternatively to returning a copy of the created resource, the location of created resource could be returned using the common:resourceReference root element (see section 5.7.5.2.2).

5.13.6 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].

5.14 Resource: Individual outbound SMS message delivery notification subscription

The resource used is:
http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/subscriptions/{subscriptionId}

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

5.14.1 Request URI variables

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serverRoot</td>
<td>server base url: hostname+port+base path. Example: <a href="http://example.com:80/ParlayREST">http://example.com:80/ParlayREST</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>version of the ParlayREST API client wants to use (e.g. 1 for version 1.x)</td>
</tr>
<tr>
<td>senderAddress</td>
<td>identifies client application. Typically SMS SHORT CODE [REST_TS_Common]</td>
</tr>
<tr>
<td>subscriptionId</td>
<td>identifier of the subscription</td>
</tr>
</tbody>
</table>

5.14.2 HTTP 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 Short Messaging, see [3GPP 29.199-4].

5.14.3 GET

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

5.14.3.1 Example (Informative)

5.14.3.1.1 Request

GET .../(apiVersion)/smsmessaging/outbound/(senderAddress)/subscriptions/(subscriptionId) 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"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/{subscriptionId}</resourceURL>
</sms:deliveryReceiptSubscription>

5.14.4 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].

5.14.5 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].

5.14.6 DELETE

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

5.14.6.1 Example (Informative)

5.14.6.1.1 Request

DELETE .../(apiVersion)/smsmessaging/outbound/(senderAddress)/subscriptions/(subscriptionId) HTTP/1.1
Accept: application/xml
Host: example.com:80
5.14.6.1.2  Response

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

5.15  Resource: Client notification about outbound SMS 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.

5.15.1  Request URI variables

Client provided.

5.15.2  HTTP Response Codes

For HTTP response codes, see [REST_TS_Common].

5.15.3  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].

5.15.4  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].

5.15.5  POST

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

5.15.5.1  Example  (Informative)

5.15.5.1.1  Request

```xml
POST /notifications/DeliveryInfoNotification 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"?>
<sms:deliveryInfoNotification xmlns:sms="urn:oma:xml:rest:sms:1">
<callbackData>12345</callbackData>
<deliveryInfo>
  <address>tel:+1350000001</address>
  <deliveryStatus>DeliveredToNetwork</deliveryStatus>
</deliveryInfo>
<link rel="DeliveryReceiptSubscription" href="http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/subscriptions/{subscriptionId}"/>
</sms:deliveryInfoNotification>
```
5.15.5.1.2 Response

HTTP/1.1 204 No Content
Content-Type: application/xml
Date: Thu, 04 Jun 2009 02:51:59 GMT

5.15.6 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

### A.1 Approved Version History

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

### A.2 Draft/Candidate Version 1.0 History

<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
</table>
| Draft Version:      | 24 Jun 2009| All      | Baseline uploaded as per agreed  
| OMA-TS-ParlayREST-ShortMessaging-V1_0 | | | OMA-ARC-REST-2009-0008-INP_ParlayREST_SMS  
|                     | 25 Nov 2009| all      | Document clean-up  
|                     | 1 Dec 2009 | App. A   | Added OMA-ARC-REST-2009-0069R02-CR_Vodafone_comments_on_SMS_API.doc  
|                     | 2 Dec 2009 | 5+       | Added OMA-ARC-REST-2009-0096-CR_Adding_Resources_to_SMS_TS  
|                     | 2 Dec 2009 | 5+       | Added OMA-ARC-REST-2009-0098R03-CR_Changes_to_SMS_API.doc and merged OMA-TS-ParlayREST_ShortMessaging-V1_0-20091112-D-changedbyCR0098R02.doc  
|                     | 3 Dec 2009 |          | Updated selfURL to resourceURL and added Link  
|                     | 3 Dec 2009 | 4        | Updated from OMA-ARC-REST-2009-0113-INP_SMS_Intro_section.  
|                     | 3 Dec 2009 | 2        | Updated from OMA-ARC-REST-2009-0114-INP_SMS_TS_Reference_Section  
|                     | 3 Dec 2009 | 3        | Updated from OMA-ARC-REST-2009-0115R01-INP_SMS_TS_Section_3.doc  
|                     | 3 Dec 2009 | 5        | Update from OMA-ARC-REST-2009-0127R01-CR_Fix_retrievalOrder,  
|                     | 7 Dec 2009 | 4, App   | Added OMA-ARC-REST-2009-0118R01-CR_Outline_for_request_response_examples.doc  
|                     | 11 Dec 2009| all      | Update after final CC, see OMA-ARC-REST-2009-0170-MINUTES_11Dec2009_CC for details  
|                     | 15 Dec 2009| all      | Last corrections added  
|                     | 16 Dec 2009| All      | Editorial fixes: styles as per template  
|                     | 26 Jan 2010| All      | CONRR editorial comments applied, G005, G007, G008, B002, B006, B042, B059, B060  
|                     | 02 Feb 2010| All      | Added OMA-ARC-REST-2010-0003R01  
|                     | 03 Feb 2010| All      | Added OMA-ARC-REST-2010-0006R02, closing B009, B010, B012  
|                     | 04 Feb 2010| All      | Added OMA-ARC-REST-2010-0007R03, closing B003, B013  
|                     | 04 Feb 2010| All      | Applied G001, G002, G004  
|                     | 09 Feb 2010| Many     | Fixed the implementation of G002 and B027 which were only partially implemented.  
<p>|                     |          |          | Clericals |</p>
<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
</table>
|                     | 22 Feb 2010| Many     | Editorial fixes  
- case convention (MESSAGE ID → messageID)  
- alignments (messageRequestId, requestId → requestId)  
- fonts, typos  
Inclusion of latest validated XML examples, closing B005, B017, B018  
Comments implemented: B004, B007, B014, B067, B068  
Comments implementation fixed and/or validated: G006, B070  
Structure of Examples sections aligned with TSs Payment and TerminalLocation (comment D018)  
History Box fixed |
|                     | 23 Feb 2010| Many     | CRs implemented  
- OMA-ARC-REST-2010-0077  
- OMA-ARC-REST-2010-0052R01 (closing B021)  
- OMA-ARC-REST-2010-0056 (closing B062)  
- OMA-ARC-REST-2010-0038R01 (closing B008) |
|                     | 24 Feb 2010| Many     | Editorial: Fixed mistake in implementation of CR 38R01  
Removed some occurrences of “online” and “offline” that were overlooked when implementing B004  
Removed some occurrences of “Request” that remained after implementing D018  
Fixed some typos |
|                     | 07 Mar 2010| Many     | CRs implemented  
OMA-ARC-REST-2010-0111R01-INP_editorial_alignments  
OMA-ARC-REST-2010-0108-CR_SMS_TS_SCR_Corrections  
OMA-ARC-REST-2010-0100-CR_Inconsistent_retrieval_DeliveryStatus_for_SMS  
OMA-ARC-REST-2010-0089-CR_Closing echoing issue examples_SMS  
OMA-ARC-REST-2010-0087R01-CR_SMS_OutboundSubscriptions_GET_missing  
OMA-ARC-REST-2010-0084-CR_Correlator_resolution_TS_SMS_updated_examples  
OMA-ARC-REST-2010-0080R01-CR_Correlator_resolution_TS_SMS  
Editorial:  
References alphabetized  
added abbreviations  
re-arranged section 5.2 such that first all types, then all enumerations are given  
removed entry “Tables” from TOC as there are no tables with captions  
Note: this revision contains changes that do require update of the XSDs. |
|                     | 10 Mar 2010| Many     | Implemented all CRs that were agreed in REST R&A on March 10  
OMA-ARC-REST-2010-0103R01-CR_More_SMS_inconsistency_fixes  
OMA-ARC-REST-2010-0101-CR_Document_74_followup_SMS  
Note: this revision contains changes that do require update of the XSDs. |
|                     | 11 Mar 2010| Many     | Editorial fixes to eradicate remains of “campaign”.  
Editorial fixes after a walk-through.  
Note that this revision contains no changes that require update of the XSDs. |
|                     | 12 Mar 2010| Many     | Further editorial fixes  
Note that this revision contains no changes that require update of the XSDs. |
<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23 Mar 2010</td>
<td>Many</td>
<td>Action resolved: AI REST-2010-A014: to REST all TS Editors: Remove the dot at the end of the resources throughout the documents. CRs implemented: OMA-ARC-REST-2010-0124-CR_TS_SMS_with_JSON_Examples OMA-ARC-REST-2010-0122-CR_Charging_Appendix_C1_SMS OMA-ARC-REST-2010-0121-CR_Fixing_wrong_example_appendix_C3_SMS Some editorial fixes. Note that this revision contains no changes that require update of the XSDs.</td>
</tr>
<tr>
<td></td>
<td>28 Mar 2010</td>
<td>Many</td>
<td>CRs implemented: OMA-ARC-REST-2010-0126R01-CR_Incorrect_URL_for_client_resources_SMS OMA-ARC-REST-2010-0155-CR_Resource_restructuring_SMS OMA-ARC-REST-2010-0135R01-CR_Correction_of_section_5.3_in_SMS_TS OMA-ARC-REST-2010-0140R01-CR_Small_corrections_and_editoral_changes_SMS Download OMA-ARC-REST-2010-0142-CR_Common_prefix_SMS OMA-ARC-REST-2010-0142-CR_Common_prefix_SMS OMA-ARC-REST-2010-0146R01-CR_PX_GetSmsDeliveryStatus_SMS OMA-ARC-REST-2010-0149-CR_Message_id_optional_TS_SMS OMA-ARC-REST-2010-0152-CR_Improvements_of_section_5.3_in_SMS_TS Some editorial fixes: Table width alignment. Fixed HTTP header in C.4.1.2 (aligned them with the XML pendant in 5.13.5.1.2) removed remaining client-related placeholders in HTTP headers for notifications [client host] etc, and ensured client host is in the notification examples always addressed as application.example.com:80 rather than example.com:80 which is the server. This is consistent with the notifyURL (<a href="http://application.example.com/notifications/DeliveryInfoNotification">http://application.example.com/notifications/DeliveryInfoNotification</a>) we use in the examples. See also item 5 in INP OMA-ARC-REST-2010-0111R01. Other: Implemented outcome of action REST-2010-A010 as communicated in eMail from Michael Brenner to ARC exploder, 29th Mar 2010, 09:52 GMT Note: this revision contains changes that do require update of the XSD.</td>
</tr>
<tr>
<td></td>
<td>29 Mar 2010</td>
<td>Many</td>
<td>Editorial: …/notifications → /notifications Note that this revision contains no changes that require update of the XSDs.</td>
</tr>
<tr>
<td>Candidate Version:</td>
<td>27 Apr 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP: OMA-TP-2010-0186-INP_ParlayREST_V1_0_ERP_forCandidate_Approval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.5.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.10.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D.17</td>
<td></td>
</tr>
<tr>
<td>Document Identifier</td>
<td>Date</td>
<td>Sections</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>13 Jul 2010</td>
<td>5.1</td>
<td>Implemented Agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.6.3.2</td>
<td>OMA-ARC-REST-2010-0314-CR_TS_SCR_changes_for_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>OMA-ARC-REST-2010-0316R01-CR_TS_SMS_remove_non_form_url_encoded_examples</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C</td>
<td>OMA-ARC-REST-2010-0330-CR_Error_404_TS_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19 Jul 2010</td>
<td>App B</td>
<td>Implemented Agreed CR:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0358-CR_Fix_SCR_errors_TS_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Candidate Version:</td>
<td>24 Aug 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP:</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST_ShortMessaging-V1_0</td>
<td></td>
<td></td>
<td>OMA-TP-2010-0359-INP_ParlyREST_V1_0_ERP_for_Candidate_reapproval</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draft Versions:</td>
<td>27 Sep 2010</td>
<td>Various</td>
<td>Implemented Agreed CRs:</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST_ShortMessaging-V1_0</td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0411-CR_Fixing_Appendic_C_Intro_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0424R02-CR_SMS_examples_for_send_failure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0426R01-CR_SMS_optional_resourceReference_in_resource_table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0434R01-CR_Fix_JSON_Examples_TS_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0450R01-CR_Fix_JSON_References_TS_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0504-CR_Closing_Actions_A63_A67_A84_A86_TS_SMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>08 Oct 2010</td>
<td>3.2, C.1</td>
<td>Implemented Agreed CR:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C.3</td>
<td>OMA-ARC-REST-2010-0560-CR_TS_SMS_small_fixes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fixed [OMADICT]</td>
</tr>
<tr>
<td></td>
<td>18 Oct 2010</td>
<td>5.1</td>
<td>Implemented Agreed CRs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.2.8</td>
<td>OMA-ARC-REST-2010-0565-CR_SMS_JSON_Update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>App D</td>
<td>OMA-ARC-REST-2010-0568-CR_SMS_more_JSON_fixes_implement_before_565</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OMA-ARC-REST-2010-0569-CR_SMS_senderAddress_attribute_description</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Candidate Version:</td>
<td>23 Nov 2010</td>
<td>All</td>
<td>Status changed to Candidate by TP:</td>
</tr>
<tr>
<td>OMA-TS-ParlayREST_ShortMessaging-V1_0</td>
<td></td>
<td></td>
<td>OMA-TP-2010-0463R01-INP_ParlayREST_V1_0_ERP_for_Candidate_reapproval</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements (Normative)

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

## B.1 SCR for ParlayREST.SMS Server

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

### B.1.1 SCR for ParlayREST.SMS.Inbound.Registration Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-INB-OFF-S-001-M</td>
<td>Support for reliable inbound message delivery</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-OFF-S-002-M</td>
<td>Retrieve messages from server - GET</td>
<td>5.4.3</td>
<td></td>
</tr>
</tbody>
</table>

### B.1.2 SCR for ParlayREST.SMS.Inbound.Registration.RetrieveDelete Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-INB-OFF-RETDEL-S-001-O</td>
<td>Support for inbound message delivery</td>
<td>5.5</td>
<td>PARLAYREST-SMS-INB-OFF-RETDEL-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-OFF-RETDEL-S-002-O</td>
<td>Retrieve messages from server - POST</td>
<td>5.5.5</td>
<td></td>
</tr>
</tbody>
</table>

### B.1.3 SCR for ParlayREST.SMS.Individual.Inbound Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-IND-INB-S-001-M</td>
<td>Support for inbound individual message delivery</td>
<td>5.6</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-IND-INB-S-002-O</td>
<td>Retrieve one message from server - GET</td>
<td>5.6.3</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-IND-INB-S-003-M</td>
<td>Confirm and delete retrieved message from server - DELETE</td>
<td>5.6.6</td>
<td></td>
</tr>
</tbody>
</table>
## B.1.4 SCR for ParlayREST.SMS.Inbound.Subscr Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-INB-ONL-SUBSCR-S-001-M</td>
<td>Support inbound subscriptions</td>
<td>5.7</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-ONL-SUBSCR-S-002-O</td>
<td>Read active subscriptions - GET</td>
<td>5.7.3</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-ONL-SUBSCR-S-003-M</td>
<td>Create inbound message subscription - POST (XML or JSON)</td>
<td>5.7.5</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-ONL-SUBSCR-S-004-O</td>
<td>Create inbound message subscription – POST (www-form-urlencoded)</td>
<td>C.3</td>
</tr>
</tbody>
</table>

## B.1.5 SCR for ParlayREST.SMS.Inbound.Individual.Subscr Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-INB-INDON-SUBSCR-S-001-M</td>
<td>Support for control and read access to individual inbound subscription</td>
<td>5.8</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-INDON-SUBSCR-S-002-O</td>
<td>Read individual inbound subscription - GET</td>
<td>5.8.3</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-INDON-SUBSCR-S-003-M</td>
<td>Update individual inbound subscriptions - DELETE</td>
<td>5.8.6</td>
</tr>
</tbody>
</table>

## B.1.6 SCR for ParlayREST.SMS.Inbound.Notifications Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-INB-NOTIF-S-001-M</td>
<td>Support for notifying application about inbound messages</td>
<td>5.9</td>
</tr>
<tr>
<td>PARLAYREST-SMS-INB-NOTIF-S-002-M</td>
<td>Notify application about inbound message arrival - POST</td>
<td>5.9.5</td>
</tr>
</tbody>
</table>

## B.1.7 SCR for ParlayREST.SMS.Outbound Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-OUTB-S-001-M</td>
<td>Support for outbound SMS messages</td>
<td>5.10</td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-S-002-O</td>
<td>Retrieve list of pending outgoing message requests - GET</td>
<td>5.10.3</td>
</tr>
<tr>
<td>PARLAYREST-SMS-</td>
<td>Create outgoing</td>
<td>5.10.5</td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>OUTB-S-003-M</td>
<td>message request - POST (XML and JSON)</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-S-004-O</td>
<td>Create outgoing message request - POST (www-form-urlencoded)</td>
<td>C.1</td>
</tr>
</tbody>
</table>

### B.1.8 SCR for ParlayREST.SMS.Outbound.MsgAndDeliveryStatus Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-OUTB-MSGDELSTAT-S-001-O</td>
<td>Support for requesting an outbound SMS message and its delivery status</td>
<td>5.11</td>
<td>PARLAYREST-SMS-OUTB-MSGDELSTAT-S-002-O</td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-MSGDELSTAT-S-002-O</td>
<td>Retrieve Outgoing Message Delivery Status - GET</td>
<td>5.11.3</td>
<td></td>
</tr>
</tbody>
</table>

### B.1.9 SCR for ParlayREST.SMS.Outbound.DeliveryStatus Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-OUTB-DELSTAT-S-001-M</td>
<td>Support for requesting delivery status of outbound SMS messages</td>
<td>5.12</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-DELSTAT-S-002-M</td>
<td>Retrieve Outgoing Message Delivery Status - GET</td>
<td>5.12.3</td>
<td></td>
</tr>
</tbody>
</table>

### B.1.10 SCR for ParlayREST.SMS.Outbound.Subscriptions Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-OUTB-SUBSCR-S-001-M</td>
<td>Support for outbound subscriptions for a particular client</td>
<td>5.13</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-SUBSCR-S-002-O</td>
<td>Read all outbound SMS delivery notification subscriptions - GET</td>
<td>5.13.3</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-SUBSCR-S-003-M</td>
<td>Create new outbound message subscription – POST (XML and JSON)</td>
<td>5.13.5</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-SUBSCR-S-004-O</td>
<td>Create new outbound message subscription – POST (www-form-urlencoded)</td>
<td>C.2</td>
<td></td>
</tr>
</tbody>
</table>
### B.1.11 SCR for ParlayREST.SMS.Individual.Outbound.Subscr Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-IND-OUTB-IND-SUBSCR-S-001-M</td>
<td>Support for outbound subscriptions for a particular client</td>
<td>5.14</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-IND-OUTB-IND-SUBSCR-S-002-O</td>
<td>Read individual SMS delivery notification subscription - GET</td>
<td>5.14.3</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-IND-OUTB-IND-SUBSCR-S-003-M</td>
<td>Delete subscription for the client - DELETE</td>
<td>5.14.6</td>
<td></td>
</tr>
</tbody>
</table>

### B.1.12 SCR for ParlayREST.SMS.Outbound.DeliveryStatus.Notifications Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARLAYREST-SMS-OUTB-DELSTAT-NOTIF-S-001-M</td>
<td>Support for notifying application about delivery status of outbound messages</td>
<td>5.15</td>
<td></td>
</tr>
<tr>
<td>PARLAYREST-SMS-OUTB-DELSTAT-NOTIF-S-002-M</td>
<td>Notify application about delivery status of outbound message - POST</td>
<td>5.15.5</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C. Application/x-www-form-urlencoded Request Format for Selected REST Operations

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

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

The following SMS REST operations are defined in this section:

- Sending a SMS to a terminal
- A mechanism to start the notification of delivery receipts
- A mechanism to start the notification of received SMS

C.1 Send a SMS to a terminal

This operation is used to create an outgoing message request.

The request parameters are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>One or more addresses to which the SMS will be sent. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '%2B' followed by the country code before the subscriber number; e.g. tel:%2B447990123456.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The address of the sender to whom a responding SMS may be sent. If the address is in the form of an MSISDN, include the protocol prefix 'tel:' and '%2B' followed by the country code before the subscriber number; e.g. tel:%2B447990123456.</td>
</tr>
<tr>
<td>message</td>
<td>xsd:string</td>
<td>No</td>
<td>The message to be sent</td>
</tr>
<tr>
<td>notifyURL</td>
<td>xsd:anyURI</td>
<td>Yes</td>
<td>URL to notify the application for delivery receipts</td>
</tr>
<tr>
<td>callbackData</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.</td>
</tr>
</tbody>
</table>
| 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} |
<p>| 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. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>senderName</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Name of the sender to appear on the terminal</td>
</tr>
<tr>
<td>chargingDescription</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Description of charge to apply to this message. In case charging is required, this parameter MUST be present.</td>
</tr>
<tr>
<td>chargingCurrency</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Currency of charge to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.</td>
</tr>
<tr>
<td>chargingAmount</td>
<td>xsd:decimal</td>
<td>Yes</td>
<td>Charging amount to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.</td>
</tr>
<tr>
<td>chargingCode</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Charging code to apply to this message. In case chargingDescription is not present, this parameter MUST NOT be present.</td>
</tr>
</tbody>
</table>

### C.1.1 Example

#### C.1.1.1 Request

```bash
POST .../{apiVersion}/smsmessaging/outbound/{senderAddress}/requests HTTP/1.1
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Accept: application/xml

address=tel:+13500000991&
address=tel:+13500000992&
senderAddress=tel:+13500000993&
message=Hello%20world&
clientCorrelator=123456&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification&
notificationFormat=XML&
senderName=Bob
```

#### C.1.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:sms:1">
  <address>tel:+13500000991</address>
  <address>tel:+13500000992</address>
  <senderAddress>tel:+13500000993</senderAddress>
  <receiptRequest>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <notificationFormat>XML</notificationFormat>
  </receiptRequest>
  <outboundSMSTextMessage>
    <message>Hello world</message>
  </outboundSMSTextMessage>
</sms:outboundSMSMessageRequest>
```
C.2 Start delivery receipt notification

This REST method is used by the application to start the delivery receipt notifications. It MUST use the HTTP POST method.

The following parameters are defined:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filterCriteria</td>
<td>xsd:string</td>
<td>No</td>
<td>Provides flexibility for the application to filter on, for example, the first 4 digits of MSISDN</td>
</tr>
<tr>
<td>notifyURL</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Notification endpoint definition</td>
</tr>
<tr>
<td>callbackData</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.</td>
</tr>
</tbody>
</table>
| 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. |

If the operation was successful, it returns an HTTP Status of “201 Created”.

C.2.1 Example (Informative)

C.2.1.1 Request

POST .../(apiVersion)/smsmessaging/outbound/subscriptions HTTP/1.1  
Host: example.com:80  
Content-Type: application/x-www-form-urlencoded  
Accept: application/xml

filterCriteria=0102&  
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification
C.2.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://{serverRoot}/(apiVersion)/smsmessaging/outbound/subscriptions/{subscriptionId}
Date: Thu, 04 Jun 2009 02:51:59 GMT

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

C.3 Start SMS notification

This REST method is used by the application to start the notification of received SMS. It MUST use the HTTP POST method.

The following parameters are defined:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Values</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destinationAddress</td>
<td>xsd:anyURI [1..unbounded]</td>
<td>No</td>
<td>Destination address of SMS</td>
</tr>
<tr>
<td>criteria</td>
<td>xsd:string</td>
<td>Yes</td>
<td>The text to match against to determine the application to receive the notification</td>
</tr>
<tr>
<td>notifyURL</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>Notification endpoint definition</td>
</tr>
<tr>
<td>callbackData</td>
<td>xsd:string</td>
<td>Yes</td>
<td>Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications.</td>
</tr>
</tbody>
</table>
| 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. |

This operation returns a result indicating whether the operation has been successful.
C.3.1 Example (Informative)

C.3.1.1 Request

Note that this example also illustrates the use of “callbackData”.

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

destinationAddress=+34680180999
criteria=Vote&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification&
callbackData=12345
```

C.3.1.2 Response

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>+34680180999</destinationAddress>
  <criteria>Vote</criteria>
  <resourceURL>http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId1}</resourceURL>
</sms: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 [RFC 4627].

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

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

D.1  Inbound message delivery (section 5.4.3.1)

Request:

```
GET .../{apiVersion}/smsmessaging/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

{"inboundSMSMessageList": {
    "inboundSMSMessage": [
        {
            "dateTime": "2009-11-19T12:00:00",
            "destinationAddress": "MSISDN1",
            "message": "First simple message",
            "messageId": "{messageId1}",
            "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1}"
        },
        {
            "dateTime": "2009-11-19T12:00:00",
            "destinationAddress": "MSISDN1",
            "message": "Second simple message",
            "messageId": "{messageId2}",
            "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId2}"
        }
    ],
    "numberOfMessagesInThisBatch": "2",
    "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages",
    "totalNumberOfPendingMessages": "20"
}}
```
D.2 maxBatchSize exceeding the allowed size (section 5.4.3.2)

Request:

GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages?maxBatchSize=5000 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}/smsmessaging/inbound/registrations/{registrationId}/messages?maxBatchSize=5000",
    "rel": "InboundSMSMessageList"
  },
  "policyException": {
    "messageId": "POL0001",
    "text": "A policy error occurred. Error code is maxBatchSize exceeded. The maximum allowed maxBatchSize is %1.",
    "variables": "20"
  }
}}

D.3 Retrieve and delete using registration (section 5.5.5.1)

Request:

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

{"inboundSMSMessageRetrieveAndDeleteRequest": {
  "maxBatchSize": "3",
  "retrievalOrder": "OldestFirst"
}}

Response:

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

{"inboundSMSMessageList": {
  "inboundSMSMessage": [
    {
      "destinationAddress": "{shortCode1}",
      "body": "Message body",
      "timestamp": "2009-06-04T02:51:59Z"
    }
  ]
}}
D.4 Inbound messages for a given registration (section 5.6.3.1)

Request:

```
GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1}?resFormat=JSON HTTP/1.1
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

{"inboundSMSMessage": {
  "dateTime": "2009-11-19T12:00:00",
  "destinationAddress": "MSISDN1",
  "message": "First simple message",
  "messageId": "{messageId1}\",
  "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1}"n",
  "senderAddress": "MSISDN2"n
}}
``` 

D.5 Invalid (non-existing) messageId (section 5.6.3.2)

Request:

```
GET .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1} HTTP/1.1
Accept: application/json
Host: example.com:80
``` 

Response:

```
HTTP/1.1 404 Not Found
```
D.6 Remove message from gateway storage (section 5.6.6.1)

Request:

```
DELETE .../{apiVersion}/smsmessaging/inbound/registrations/{registrationId}/messages/{messageId1} 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.7 Read active subscriptions (section 5.7.3.1)

Request:

```
GET .../{apiVersion}/smsmessaging/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"
  },
  "criteria": "Urgent*",
...}
```
"destinationAddress": "680180999",
"resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId1}"
},
{
"callbackReference": {
"callbackData": "54321",
"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
},
"criteria": "Urgent*",
"destinationAddress": "80999",
"resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId2}"
}
]]

D.8 Create new message subscription (section 5.7.5.1)

Request:

POST .../{apiVersion}/smsmessaging/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"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://{serverRoot}/{apiVersion}/smsmessaging/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"
},
"criteria": "Urgent*",
"destinationAddress": "+34680180999",
"resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId1}"
}}
D.9 Returning the location of created resource (section 5.7.5.2)

Request:

POST .../{apiVersion}/smsmessaging/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"
}}

Response:

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

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

D.10 Read individual subscription (section 5.8.3.1)

Request:

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

Response:

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

{"subscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
   },
   "criteria": "Urgent*",
   "destinationAddress": "+34680180999",
   "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/inbound/subscriptions/{subscriptionId}"}}
D.11 Delete a subscription (section 5.8.6.1)

Request:

DELETE .../{apiVersion}/smsmessaging/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.12 Notify client about message arrival (section 5.9.5.1)

Request:

POST /notifications/DeliveryInfoNotification HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Content-Length: nnnn
Host: application.example.com:80

{"inboundSMSMessageNotification": {
    "callbackData": "12345",
    "inboundSMSMessage": {
        "dateTime": "2009-11-19T12:00:00",
        "destinationAddress": "MSISDN",
        "message": "First simple message",
        "messageId": "{messageId1}"
        "senderAddress": "sender name"
     }
}}

Response:

HTTP/1.1 204 No Content
Content-Type: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT

D.13 Retrieve list of pending outbound messages (section 5.10.3.1)

Request:

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

Response:
D.14 Create outbound message, returning a representation of created resource (section 5.10.5.1)

Request:

POST .../{apiVersion}/smsmessaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80
D.15 Create outbound message, returning the location of created resource (section 5.10.5.2)

Request:

POST .../{apiVersion}/smsmessaging/outbound/{senderAddress}/requests HTTP/1.1
Accept: application/json
D.16 ServiceException in case of single address or all multiple addresses failure (section 5.10.5.3)

Request:

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

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+1350000001",
    "tel:+1350000999"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "senderAddress": "tel:+1351111999",
  "senderName": "MyName"
}}

Response:

HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: nnn
Date: Thu, 04 Jun 2009 02:51:59 GMT
D.17  Multiple addresses partial success, with deliveryInfoList in response (section 5.10.5.4)

Request:

POST .../(apiVersion)/smsmessaging/outbound/(senderAddress)/requests HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=UTF-8
Host: example.com:80

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+1350000001",
    "tel:+1350000999"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"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}/smsmessaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: nnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+1350000001",
    "tel:+1350000999"
  ],
  "clientCorrelator": "67893",
  "deliveryInfoList": {
    "deliveryInfo": [
      {"address": "tel:+1350000001","deliveryStatus": "MessageWaiting"},
      {"address": "tel:+1350000999","deliveryStatus": "DeliveryImpossible"}
    ]
  }
}
### D.18 Multiple addresses partial success, without deliveryInfoList in response (section 5.10.5.5)

**Request:**

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

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+1350000001",
    "tel:+1350000999"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"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}/smsmessaging/outbound/{senderAddress}/requests/{requestId}
Content-Length: nnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+1350000001",
    "tel:+1350000999"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "senderAddress": "tel:+1351111999",
  "senderName": "MyName"
}}
```
D.19 Get message delivery status (section 5.11.3.1)

Request:

GET …/{apiVersion}/smsmessaging/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

{"outboundSMSMessageRequest": {
"address": [ 
  "tel:+1350000001",
  "tel:+1350000999"
],
"clientCorrelator": "67893",
"deliveryInfoList": {
  "deliveryInfo": [
    {
      "address": "tel:+1350000001",
      "deliveryStatus": "MessageWaiting"
    },
    {
      "address": "tel:+1350000999",
      "deliveryStatus": "MessageWaiting"
    }
  ],
  "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}/deliveryInfos"
},
"outboundSMSTextMessage": {"message": "sent message"},
"receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
"resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/requests/{requestId}"
"senderAddress": "tel:+1351111999",
"senderName": "MyName"
}}

D.20 Get message delivery status (section 5.12.3.1)

Request:

GET …/{apiVersion}/smsmessaging/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
Content-Length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document
D.21 Read delivery notification subscriptions (section 5.13.3.1)

Request:

GET .../{apiVersion}/smsmessaging/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}/smsmessaging/outbound/subscriptions/{subscriptionId}"
    },
    {
      "callbackReference": {
        "callbackData": "54321",
        "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
      },
      "filterCriteria": "0103",
      "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/{subscriptionId}"
    }
  ],
  "resourceURL": "http://{serverRoot}/{apiVersion}/smsmessaging/outbound/subscriptions/"
}}

D.22 Create delivery notification subscription (section 5.13.5.1)

Request:
D.23 Read delivery notification subscription (section 5.14.3.1)

Request:

```
GET .../{apiVersion}/smsmessaging/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}/smsmessaging/outbound/subscriptions/{subscriptionId}"
}}
```
D.24 Delete subscription for a client (section 5.14.6.1)

Request:

DELETE .../{apiVersion}/smsmessaging/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.25 Notify client about message delivery status (section 5.15.5.1)

Request:

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

{"deliveryInfoNotification": {
  "callbackData": "12345",
  "deliveryInfo": {
    "address": "tel:+1350000001",
    "deliveryStatus": "DeliveredToNetwork"
  },
  "link": {
    "href": "http://{serverRoot}/{apiVersion}/smsmessaging/outbound/{senderAddress}/subscriptions/{subscriptionId}"
  }
}}

Response:

HTTP/1.1 204 No Content
Content-Type: application/json
Date: Thu, 04 Jun 2009 02:51:59 GMT
**Appendix E. Parlay X operations mapping**

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

<table>
<thead>
<tr>
<th>ParlayREST Resource</th>
<th>ParlayREST Method</th>
<th>ParlayREST Section reference</th>
<th>Parlay X equivalent operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound SMS message requests for a given registration</td>
<td>GET</td>
<td>5.4.3</td>
<td>getReceivedSms</td>
</tr>
<tr>
<td>Inbound SMS messages Retrieve and Delete using registration</td>
<td>POST</td>
<td>5.5.5</td>
<td>getReceivedSms</td>
</tr>
<tr>
<td>Inbound SMS message subscriptions</td>
<td>POST</td>
<td>5.7.5</td>
<td>startSmsNotification</td>
</tr>
<tr>
<td>Individual inbound SMS message subscription</td>
<td>DELETE</td>
<td>5.8.6</td>
<td>stopSmsNotification</td>
</tr>
<tr>
<td>Client notification about inbound SMS message</td>
<td>POST</td>
<td>5.9.5</td>
<td>notifySmsReception</td>
</tr>
<tr>
<td>Outbound SMS message requests</td>
<td>POST</td>
<td>5.10.5</td>
<td>sendSmsRequest</td>
</tr>
<tr>
<td>Outbound SMS message delivery status</td>
<td>GET</td>
<td>5.12.3</td>
<td>getSmsDeliveryStatus</td>
</tr>
<tr>
<td>Outbound SMS message delivery notification subscriptions</td>
<td>POST</td>
<td>5.13.5</td>
<td>startDeliveryReceiptNotification</td>
</tr>
<tr>
<td>Individual outbound SMS message delivery notification subscription</td>
<td>DELETE</td>
<td>5.14.6</td>
<td>stopDeliveryReceiptNotification</td>
</tr>
<tr>
<td>Client notification about outbound SMS message delivery status</td>
<td>POST</td>
<td>5.15.5</td>
<td>notifySmsDeliveryReceipt</td>
</tr>
</tbody>
</table>

1) Note: The ParlayX SOAP operation getReceivedSms is similar to but not quite the same as this ParlayREST method because DELETE of individual message is required for confirmation of successful retrieval (see DELETE on Inbound SMS message).