## Contents

1. SCOPE ................................................................................................................................................. 9
2. REFERENCES ........................................................................................................................................... 10
   2.1 NORMATIVE REFERENCES ........................................................................................................... 10
   2.2 INFORMATIVE REFERENCES ....................................................................................................... 10
3. TERMINOLOGY AND CONVENTIONS ................................................................................................. 11
   3.1 CONVENTIONS ............................................................................................................................. 11
   3.2 DEFINITIONS ................................................................................................................................... 11
   3.3 ABBREVIATIONS .......................................................................................................................... 11
4. INTRODUCTION ................................................................................................................................... 13
   4.1 VERSION 1.0 ................................................................................................................................... 13
5. SHORT MESSAGING SERVICE (SMS) API DEFINITION ........................................................................ 14
   5.1 RESOURCES SUMMARY .................................................................................................................. 14
   5.2 DATA TYPES ................................................................................................................................. 21
   5.2.1 XML Namespaces ..................................................................................................................... 21
   5.2.2 Structures ............................................................................................................................... 21
   5.2.2.1 Type: InboundSMSMessageList .......................................................................................... 21
   5.2.2.2 Type: InboundSMSMessage ............................................................................................... 21
   5.2.2.3 Type: InboundSMSMessageNotification .......................................................................... 22
   5.2.2.4 Type: SubscriptionList ........................................................................................................ 23
   5.2.2.5 Type: Subscription ................................................................................................................ 23
   5.2.2.6 Type: InboundSMSMessageRetrieveAndDeleteRequest ..................................................... 24
   5.2.2.7 Type: OutboundSMSMessageRequestList .......................................................................... 24
   5.2.2.8 Type: OutboundSMSMessageRequest ................................................................................. 25
   5.2.2.9 Type: OutboundSMSTextMessage ...................................................................................... 26
   5.2.2.10 Type: OutboundSMSBinaryMessage ................................................................................ 27
   5.2.2.11 Type: OutboundSMSTextLogoMessage ............................................................................. 27
   5.2.2.12 Type: OutboundSMSRingToneMessage ........................................................................... 27
   5.2.2.13 Type: OutboundSMSTextLogoFlashMessage ...................................................................... 27
   5.2.2.14 Type: DeliveryInfoList ....................................................................................................... 27
   5.2.2.15 Type: DeliveryInfoNotification ........................................................................................ 28
   5.2.2.16 Type: DeliveryInfo ............................................................................................................... 28
   5.2.2.17 Type: DeliveryReceiptSubscriptionList ............................................................................ 29
   5.2.2.18 Type: DeliveryReceiptSubscription .................................................................................. 29
   5.2.3 Enumerations ............................................................................................................................. 30
   5.2.3.1 Enumeration: DeliveryStatus .............................................................................................. 30
   5.2.3.2 Enumeration: SmsFormat .................................................................................................... 30
   5.2.3.3 Enumeration: RetrievalOrder .............................................................................................. 31
   5.2.4 Values of the Link “rel” attribute ............................................................................................. 31
5.3 SEQUENCE DIAGRAMS ...................................................................................................................... 31
   5.3.1 Send SMS and check the delivery status .................................................................................... 31
   5.3.2 Inbound SMS message delivery (push mode) .......................................................................... 32
   5.3.3 Inbound SMS message delivery (polling mode) ....................................................................... 33
6. DETAILED SPECIFICATION OF THE RESOURCES ............................................................................ 35
   6.1 RESOURCE: INBOUND SMS MESSAGE REQUESTS FOR A GIVEN REGISTRATION ................. 35
   6.1.1 Request URL variables .............................................................................................................. 35
   6.1.2 Response Codes and Error Handling ....................................................................................... 36
   6.1.3 GET .............................................................................................................................................. 36
   6.1.3.1 Example 1: Inbound message delivery (Informative) .......................................................... 36
   6.1.3.1.1 Request ........................................................................................................................... 36
   6.1.3.1.2 Response ......................................................................................................................... 36
   6.1.3.2 Example 2: maxBatchSize exceeding the allowed size (Informative) ................................... 37
   6.1.3.2.1 Request ........................................................................................................................... 37
   6.1.3.2.2 Response ......................................................................................................................... 37
   6.1.4 PUT ............................................................................................................................................... 37
   6.1.5 POST ............................................................................................................................................. 37
6.2 Resource: Inbound SMS messages retrieve and delete using registration ......................................................... 38

6.2.1 Request URL variables ................................................................................................................................. 38
6.2.2 Response Codes and Error Handling ............................................................................................................ 38
6.2.3 GET .............................................................................................................................................................. 38
6.2.4 PUT .............................................................................................................................................................. 38
6.2.5 POST .......................................................................................................................................................... 38

6.2.5.1 Example: Retrieve and delete using registration (Informative) ................................................................. 39
6.2.5.1.1 Request .................................................................................................................................................. 39
6.2.5.1.2 Response ............................................................................................................................................. 39

6.2.6 DELETE .................................................................................................................................................... 40

6.3 Resource: Inbound SMS message for a given registration .............................................................................. 40

6.3.1 Request URL variables .................................................................................................................................. 40
6.3.2 Response Codes and Error Handling ........................................................................................................... 40
6.3.3 GET .......................................................................................................................................................... 40

6.3.3.1 Example 1: Inbound messages for a given registration (Informative) ......................................................... 40
6.3.3.1.1 Request ................................................................................................................................................ 40
6.3.3.1.2 Response ............................................................................................................................................ 40

6.3.6 DELETE ................................................................................................................................................... 42

6.3.6.1 Example: Remove message from gateway storage (Informative) ............................................................ 42
6.3.6.1.1 Request ............................................................................................................................................... 42
6.3.6.1.2 Response ........................................................................................................................................... 42

6.4 Resource: Inbound SMS message subscriptions .......................................................................................... 42

6.4.1 Request URL variables .................................................................................................................................. 42
6.4.2 Response Codes and Error Handling ........................................................................................................... 42
6.4.3 GET .......................................................................................................................................................... 43

6.4.3.1 Example: Read active subscriptions (Informative) .................................................................................... 43
6.4.3.1.1 Request ............................................................................................................................................... 43
6.4.3.1.2 Response ............................................................................................................................................ 43

6.4.4 PUT .......................................................................................................................................................... 43

6.4.5 POST ........................................................................................................................................................ 43

6.4.5.1 Example 1: Create inbound SMS message subscription, returning a representation of created resource (Informative) ........................................................................................................................................ 44
6.4.5.1.1 Request ............................................................................................................................................... 44
6.4.5.1.2 Response ............................................................................................................................................ 44

6.4.5.2 Example 2: Create inbound SMS message subscription, returning the location of created resource (Informative) ........................................................................................................................................ 45
6.4.5.2.1 Request ............................................................................................................................................... 45
6.4.5.2.2 Response ............................................................................................................................................ 45

6.4.6 DELETE ................................................................................................................................................... 45

6.5 Resource: Individual inbound SMS message subscription ............................................................................ 45

6.5.1 Request URL variables .................................................................................................................................. 45
6.5.2 Response Codes and Error Handling ........................................................................................................... 46
6.5.3 GET .......................................................................................................................................................... 46

6.5.3.1 Example: Read individual subscription (Informative) .............................................................................. 46
6.5.3.1.1 Request ............................................................................................................................................... 46
6.5.3.1.2 Response ............................................................................................................................................ 46

6.5.4 PUT .......................................................................................................................................................... 46

6.5.5 POST ........................................................................................................................................................ 47

6.5.6 DELETE ................................................................................................................................................... 47

6.5.6.1 Example: Delete subscription (Informative) ............................................................................................. 47
6.5.6.1.1 Request ............................................................................................................................................... 47
6.5.6.1.2 Response ............................................................................................................................................ 47

6.6 Resource: Client notification about inbound SMS message ........................................................................... 47

6.6.1 Request URL variables .................................................................................................................................. 47
6.6.2 Response Codes and Error Handling ........................................................................................................... 47
6.6.3 GET .......................................................................................................................................................... 47
6.10 Resource: Outbound SMS message delivery notification subscriptions .......................................................... 60
6.10.1 Request URL variables ......................................................................................................................................... 60
6.10.2 Response Codes and Error Handling .................................................................................................................. 60
6.10.3 GET....................................................................................................................................................................... 60

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document

[OMA-TEMPLATE-TS_RESTful_Network_API-20130226-I]
6.12 RESOURCE: CLIENT NOTIFICATION ABOUT OUTBOUND SMS MESSAGE DELIVERY STATUS .................................................. 65
6.12.1 Request URL variables .................................................................................................................................................. 65
6.12.2 Response Codes and Error Handling .................................................................................................................................................. 66
6.12.3 GET........................................................................................................................................................................... 66
6.12.4 PUT........................................................................................................................................................................... 66
6.12.5 POST........................................................................................................................................................................... 66
6.12.6 DELETE........................................................................................................................................................................... 66
6.12.5.1 Example: Notify client about message delivery status (Informative) .............................................................................. 66
6.12.5.1.1 Request ........................................................................................................................................................................... 66
6.12.5.1.2 Response ........................................................................................................................................................................... 66
6.12.6 DELETE........................................................................................................................................................................... 66

7. FAULT DEFINITIONS .............................................................................................................................................................. 67
7.1 SERVICE EXCEPTIONS .............................................................................................................................................................. 67
7.1.1 SVC0280: Message too long ........................................................................................................................................... 67
7.1.2 SVC0281: Unrecognized data format .............................................................................................................................. 67
7.1.3 SVC0283: Delivery Receipt Notification not supported ........................................................................................................... 67
7.2 POLICY EXCEPTIONS .............................................................................................................................................................. 67
7.2.1 POL0109: Binary SMS not allowed ........................................................................................................................................... 68
7.2.2 POL0102: MaxBatchSize exceeded ........................................................................................................................................... 68

APPENDIX A. CHANGE HISTORY (INFORMATIVE) ........................................................................................................................... 69
A.1 APPROVED VERSION HISTORY .................................................................................................................................................. 69
A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY ...................................................................................................................................... 69

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ........................................................................... 72
B.1 SCR FOR REST.SMS SERVER .................................................................................................................................................. 72
B.1.1 SCR for REST.SMS.Inbound.Registration Server .................................................................................................................. 72
B.1.2 SCR for REST.SMS.Inbound.Registration.RetrieveDelete Server .......................................................................................... 72
B.1.3 SCR for REST.SMS.Inbound.Individual.Inbound Server ........................................................................................................... 72
B.1.4 SCR for REST.SMS.Inbound.Subscr Server ...................................................................................................................................... 73
B.1.5 SCR for REST.SMS.Individual.Inbound.Subscr Server ........................................................................................................... 73
B.1.6 SCR for REST.SMS.Inbound.Notifications Server ...................................................................................................................................... 73
B.1.7 SCR for REST.SMS.Outbound Server ...................................................................................................................................... 73
B.1.8 SCR for REST.SMS.Outbound.MsgAndDeliveryStatus Server .......................................................................................... 74
B.1.9 SCR for REST.SMS.Outbound.DeliveryStatus Server ...................................................................................................................................... 74
B.1.10 SCR for REST.SMS.Outbound.Subscriptions Server ................................................................. 74
B.1.11 SCR for REST.SMS.Individual.Outbound.Subscr Server ......................................................... 75
B.1.12 SCR for REST.SMS.Outbound.DeliveryStatus.Notifications Server ........................................ 75

APPENDIX C. APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE) ............................................................................................................ 76

C.1 SEND A SMS TO A TERMINAL........................................................................................................... 76
C.1.1 Example: Create outbound message request, returning representation of created resource in response (Informative) ........................................................................................................ 77
   C.1.1.1 Request .......................................................................................................................................... 77
   C.1.1.2 Response ....................................................................................................................................... 77

C.2 START DELIVERY RECEIPT NOTIFICATION ................................................................................. 77
C.2.1 Example: Create outbound delivery notification subscription, using tel URI (Informative) ................................................................................................................................. 79
   C.2.1.1 Request .......................................................................................................................................... 79
   C.2.1.2 Response ....................................................................................................................................... 79
C.2.2 Example: Create outbound delivery notification subscription, using ACR (Informative) ................................................................................................................................. 79
   C.2.2.1 Request .......................................................................................................................................... 79
   C.2.2.2 Response ....................................................................................................................................... 80

C.3 START SMS NOTIFICATION ...................................................................................................... 80
C.3.1 Example: Create inbound SMS message subscription, returning a representation of created resource (Informative) ........................................................................................................ 81
   C.3.1.1 Request .......................................................................................................................................... 81
   C.3.1.2 Response ....................................................................................................................................... 81

APPENDIX D. JSON EXAMPLES (INFORMATIVE) ............................................................................... 82

D.1 INBOUND MESSAGE DELIVERY (SECTION 6.1.3.1) ........................................................................ 82
D.2 MAXBATCHSIZE EXCEEDING THE ALLOWED SIZE (SECTION 6.1.3.2) .................................................. 83
D.3 RETRIEVE AND DELETE USING REGISTRATION (SECTION 6.2.5.1) ............................................... 83
D.4 INBOUND MESSAGES FOR A GIVEN REGISTRATION (SECTION 6.3.3.1) ........................................... 84
D.5 INVALID (NON-EXISTING) MESSAGEID (SECTION 6.3.3.2) ............................................................... 84
D.6 REMOVE MESSAGE FROM GATEWAY STORAGE (SECTION 6.3.6.1) .................................................... 85
D.7 READ ACTIVE SUBSCRIPTIONS (SECTION 6.4.3.1) ............................................................................ 85
D.8 CREATE INBOUND SMS MESSAGE SUBSCRIPTION, RETURNING A REPRESENTATION OF CREATED RESOURCE (SECTION 6.4.5.1) ........................................................................................................ 86
D.9 CREATE INBOUND SMS MESSAGE SUBSCRIPTION, RETURNING THE LOCATION OF CREATED RESOURCE (SECTION 6.4.5.2) ........................................................................................................ 87
D.10 READ INDIVIDUAL SUBSCRIPTION (SECTION 6.5.3.1) ....................................................................... 87
D.11 DELETE A SUBSCRIPTION (SECTION 6.5.6.1) ..................................................................................... 88
D.12 NOTIFY CLIENT ABOUT MESSAGE Arrival (SECTION 6.6.5.1) ............................................................ 88
D.13 RETRIEVE LIST OF PENDING OUTBOUND MESSAGES (SECTION 6.7.3.1) ...................................... 89
D.14 CREATE OUTBOUND MESSAGE REQUEST, RETURNING A REPRESENTATION OF CREATED RESOURCE IN RESPONSE (SECTION 6.7.5.1) ........................................................................................................ 90
D.15 CREATE OUTBOUND MESSAGE REQUEST, RETURNING THE LOCATION OF CREATED RESOURCE IN RESPONSE (SECTION 6.7.5.2) ........................................................................................................ 91
D.16 SERVICEEXCEPTION IN CASE OF SINGLE ADDRESS OR ALL MULTIPLE ADDRESSES FAILURE (SECTION 6.7.5.3) .................................................. 92
D.17 MULTIPLE ADDRESSES PARTIAL SUCCESS, WITH DELIVERYINFOList IN RESPONSE (SECTION 6.7.5.4) ........................................................................................................ 92
D.18 MULTIPLE ADDRESSES PARTIAL SUCCESS, WITHOUT DELIVERYINFOList IN RESPONSE (SECTION 6.7.5.5) ........................................................................................................ 93
D.19 CREATE OUTBOUND MESSAGE USING SHORT CODE AS SENDERADDRESS, RETURNING A REPRESENTATION OF CREATED RESOURCE (SECTION 6.7.5.6) ........................................................................................................ 94
D.20 GET MESSAGE DELIVERY STATUS (SECTION 6.8.3.1) ....................................................................... 95
D.21 GET MESSAGE DELIVERY STATUS (SECTION 6.9.3.1) ....................................................................... 96
D.22 READ DELIVERY NOTIFICATION SUBSCRIPTIONS (SECTION 6.10.3.1) ............................................. 97
D.23 CREATE OUTBOUND DELIVERY NOTIFICATION SUBSCRIPTION, USING tel URI (SECTION 6.10.5.1) ................................................................................................. 97
D.24 CREATE OUTBOUND DELIVERY NOTIFICATION SUBSCRIPTION, USING ACR (SECTION 6.10.5.2) .... 98
D.25 READ DELIVERY NOTIFICATION SUBSCRIPTION (SECTION 6.11.3.1) ..................................................... 99
D.26 DELETE SUBSCRIPTION FOR A CLIENT (SECTION 6.11.6.1) ............................................................... 99
D.27 NOTIFY CLIENT ABOUT MESSAGE DELIVERY STATUS (SECTION 6.12.5.1) ....................................... 99

APPENDIX E. PARLAY X OPERATIONS MAPPING (INFORMATIVE) ........................................... 101
APPENDIX F. LIGHT-WEIGHT RESOURCES (INFORMATIVE) ................................................................. 102

APPENDIX G. AUTHORIZATION ASPECTS (NORMATIVE) ................................................................. 103

G.1 USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIs .................................................. 103
G.1.1 Scope values ........................................................................................................................................... 103
G.1.1.1 Definitions ........................................................................................................................................... 103
G.1.1.2 Downscoping ....................................................................................................................................... 103
G.1.1.3 Mapping with resources and methods .............................................................................................. 104
G.1.2 Use of ‘acr:auth’ ..................................................................................................................................... 107

Figures

Figure 1: Resource structure defined by this specification .................................................................................. 15
Figure 2: Send SMS and check the delivery status ............................................................................................... 32
Figure 3: Inbound SMS message delivery (push mode) .......................................................................................... 33
Figure 4: Inbound SMS message delivery (polling mode) ....................................................................................... 34

Tables

Table 1: Parlay X operations mapping ................................................................................................................. 101
Table 2: Scope values for RESTful Short Messaging API ...................................................................................... 103
Table 3: Required scope values for: Inbound SMS messages for periodic polling .............................................. 105
Table 4: Required scope values for: Subscription management for inbound SMS messages .................................. 105
Table 5: Required scope values for: Sending SMS message and obtaining the delivery status .................................. 106
Table 6: Required scope values for: Subscription management for outbound SMS message delivery status .......... 106
1. Scope

This specification defines a RESTful API for Short Messaging using an HTTP protocol binding, based on the similar API defined in [3GPP 29.199-4].
2. References

2.1 Normative References


[REST_NetAPI_Common] “Common definitions for OMA RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL: http://www.openmobilealliance.org/


[REST_SUP_SMS] “XML schema for the RESTful Network API for SMS”, Open Mobile Alliance™, OMA-SUP-XMLSchema-V1.0, URL: http://www.openmobilealliance.org/


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].

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

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACR</td>
<td>Anonymous Customer Reference</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>EMS</td>
<td>Enhanced Message Service</td>
</tr>
<tr>
<td>GIF</td>
<td>Graphics Interchange Format</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>JPEG</td>
<td>Joint Photographic Expert Group</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</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>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>REST</td>
<td>REpresentational State Transfer</td>
</tr>
<tr>
<td>RTX</td>
<td>Ring Tone eXtended</td>
</tr>
<tr>
<td>SCR</td>
<td>Static Conformance Requirements</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SMSC</td>
<td>Short Message Service Center</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SMPP</td>
<td>Short Message Peer-to-Peer</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>WP</td>
<td>White Paper</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 Technical Specification for the RESTful Network API for Short Messaging contains the HTTP protocol binding based on Parlay X Short Messaging Web Services [3GPP 29.199-4] specification, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON, and application/x-www-form-urlencoded).

4.1 Version 1.0

The RESTful Network API for Short Messaging V1.0 is a republication of the ParlayREST ShortMessaging API V 1.1 [ParlayREST_SMS] as part of the suite of OMA RESTful Network APIs. Bug fixes and structural changes to fit that suite, but also functional changes have been applied.

Version 1.0 of the RESTful Network API for Short Messaging keeps supporting 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 method)

The following new functionality has been introduced:

- Support for scope values used with authorization framework defined in [Autho4API_10]
- Support for Anonymous Customer Reference (ACR) as an end user identifier
- Support for “acr:auth” as a reserved keyword in a resource URL variable that identifies an end user

All changes are backwards-compatible with ParlayREST ShortMessaging V 1.1, with the following exceptions:

- the introduction of “messages” in the resource URL path for the resource “Inbound SMS messages retrieve and delete using registration” is not backwards-compatible.
5. Short Messaging Service (SMS) API definition

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

The remainder of this document is structured as follows:

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

Section 6 contains the detailed specification for each of the resources. Each such subsection defines the resource, the request URL 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 6 use XML as the format for the message body. Application/x-www-form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix D.

Section 7 contains fault definition details such as Service Exceptions and Policy Exceptions. Appendix B provides the Static Conformance Requirements (SCR).

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

Appendix F provides a list of all light-weight resources, where applicable. Appendix G defines authorization aspects to control access to the resources defined in this specification.

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

5.1 Resources Summary

This section summarizes all the resources used by the RESTful Network API for SMS.

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

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.
Figure 1: Resource structure defined by this specification
The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

**Purpose:** To allow client to periodically poll for inbound messages (based on a provisioning step configuration)

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound SMS message requests for a given registration</td>
<td>/inbound/registrations/{registrationId}/messages</td>
<td>InboundSMSMessageList</td>
<td>GET: no, PUT: no, POST: no, DELETE: no</td>
</tr>
<tr>
<td>Inbound SMS message retrieve and delete using registration</td>
<td>/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages</td>
<td>InboundSMSMessageList (used for POST response) InboundSMSMessageRetrieveAndDeleteRequest (used for POST request)</td>
<td>GET: no, PUT: no, POST: Pops one or more messages from the gateway storage (removes it if successful), DELETE: no</td>
</tr>
<tr>
<td>Resource</td>
<td>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</td>
<td>Data Structures</td>
<td>HTTP verbs</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Inbound SMS message for a given registration</td>
<td>/inbound/registrations/{registrationId}/messages/{messageId}</td>
<td>InboundSMSMessage</td>
<td>GET: Read one message from gateway storage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT: no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POST: no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE: Delete one message from gateway storage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: Messages are automatically deleted after a certain time.</td>
</tr>
</tbody>
</table>

Purpose: To allow client to manage subscriptions for inbound messages

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound SMS message subscriptions</td>
<td>/inbound/subscriptions</td>
<td>SubscriptionList (used for GET)</td>
<td>GET: Read all active subscriptions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subscription (used for POST)</td>
<td>PUT: no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>common:ResourceReference (optional alternative for POST response)</td>
<td>POST: Create new message subscription</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE: no</td>
</tr>
<tr>
<td>Resource</td>
<td>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</td>
<td>Data Structures</td>
<td>HTTP verbs</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>Individual inbound SMS message subscription</td>
<td>/inbound/subscriptions/{subscriptionId}</td>
<td>Subscription</td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td>Purpose: To allow server to notify client about inbound messages</td>
<td></td>
<td></td>
<td>DELETE</td>
</tr>
<tr>
<td>Client notification about inbound SMS message</td>
<td>&lt;specified by the client when subscription is created or during provisioning process&gt;</td>
<td>InboundSMSMessageNotification</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>no</td>
</tr>
<tr>
<td>Purpose: To allow client to send messages and obtain delivery status for messages</td>
<td></td>
<td></td>
<td>no</td>
</tr>
</tbody>
</table>

© 2013 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
| Outbound SMS message requests | /outbound/{senderAddress}/requests | OutboundSMSMessageRequestList (used for GET)  
|                             |                                     | OutboundSMSMessageRequest (used for POST)  
|                             |                                     | common:ResourceReference (optional alternative for POST response)  
|                             |                                     | Read all pending outbound message requests  
|                             |                                     | no  
|                             |                                     | Create new outbound messages request  
|                             |                                     | no  

| Outbound SMS message request and delivery status | /outbound/{senderAddress}/requests/{requestId} | OutboundSMSMessageRequest  
|                                                 |                                         | Read a certain sent SMS message, including the deliveryStatus  
|                                                 |                                         | no  
|                                                 |                                         | no  
|                                                 |                                         | no  

| Outbound SMS message delivery status | /outbound/{senderAddress}/subscriptions | DeliveryInfoList  
|                                     |                                         | Read delivery status for the individual outbound message  
|                                     |                                         | no  
|                                     |                                         | no  
|                                     |                                         | no  

Purpose: To allow client to manage subscriptions for outbound message delivery status

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELETE</td>
</tr>
</tbody>
</table>

Outbound SMS message delivery notification subscriptions  
/outbound/{senderAddress}/subscriptions  
DeliveryReceiptSubscriptionList (used for GET)  
DeliveryReceiptSubscription (used for POST)  
common:ResourceReference (optional alternative for POST response)  
Read all outbound SMS subscriptions  
no  
Create new delivery receipt subscription  
no
<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Data Structures</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET PUT POST DELETE</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</td>
</tr>
<tr>
<td>Purpose: To allow server to notify client about outbound message delivery status</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>&lt;specified by the client&gt;</td>
<td>DeliveryInfoNotification</td>
<td>no</td>
</tr>
<tr>
<td>Client notification about outbound SMS message delivery status</td>
<td>&lt;specified by the client when outbound request is submitted&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.2 Data Types

5.2.1 XML Namespaces

The namespace for the ShortMessaging data types is:

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

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

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

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

5.2.2 Structures

The subsections of this section define the data structures used in the Short Messaging API.

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

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

5.2.2.1 Type: InboundSMSMessageList

List of inbound SMS messages.

<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 response bodies.

5.2.2.2 Type: InboundSMSMessage

Individual inbound SMS message.
## 5.2.2.3 Type: InboundSMSMessageNotification

Notification about an inbound SMS message.

<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>The ‘callbackData’ element if it was passed by the application in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>‘callbackReference’ element when creating a subscription to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>notifications about inbound SMS messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See [REST_NetAPI_Common] for details.</td>
</tr>
</tbody>
</table>

A root element named inboundSMSMessage of type InboundSMSMessage is allowed in request and/or response bodies.
<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inboundSMSMessage</td>
<td>InboundSMSMessage</td>
<td>No</td>
<td>Inbound SMS message</td>
</tr>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
<td>Link to other resources. For example a link to the original outbound message request.</td>
</tr>
</tbody>
</table>

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

### 5.2.2.4 Type: SubscriptionList

List of subscriptions to notifications about inbound SMS messages.

<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>No</td>
<td>Self referring URL</td>
</tr>
</tbody>
</table>

A root element named subscriptionList of type SubscriptionList is allowed in response bodies.

### 5.2.2.5 Type: Subscription

Individual subscription to notifications about inbound SMS messages.

<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 endpoint and parameters</td>
</tr>
<tr>
<td>destinationAddress</td>
<td>xsd:anyURI [1..unbounded]</td>
<td>No</td>
<td>The destination address of the message (e.g. 'sip' URI, 'tel' URI, 'acr' URI)</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></td>
<td></td>
<td></td>
<td>This text is matched against the first word, 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 can use to tag this particular resource representation during a request to create a resource on the server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This element MAY be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the field is present, the</td>
</tr>
</tbody>
</table>
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. The resourceURL SHALL NOT be included in POST requests by the client, but POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests. |
| link              | common:Link[0..unbounded] | Yes | Links to other resources that are in relationship with the resource |

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

5.2.2.6 Type: InboundSMSMessageRetrieveAndDeleteRequest

Parameters of the request to retrieve and delete SMS messages in one operation.

<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 than 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.

5.2.2.7 Type: OutboundSMSMessageRequestList

List of outbound SMS message requests.

<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 application and still exist in the server. Messages exist in the server for a little time after reaching their final delivery status.</td>
</tr>
</tbody>
</table>
A root element named `outboundSMSMessageRequestList` of type `OutboundSMSMessageRequestList` is allowed in response bodies.

## 5.2.2.8 Type: OutboundSMSMessageRequest

Individual outbound SMS message request.

<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 (e.g. 'sip' URI, 'tel' URI, 'acr' URI)</td>
</tr>
<tr>
<td>senderAddress</td>
<td>xsd:anyURI</td>
<td>No</td>
<td>The address of the sender to whom a responding message may be sent (e.g. 'sip' URI, 'tel' URI, 'acr' URI). If <code>senderAddress</code> is also part of the request URL, the two MUST have the same value.</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 are assumed to be set during provisioning of 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 notification endpoint and parameters that will be used to notify the application when the message has been delivered to terminal or if delivery is impossible.</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>outboundSMSFlashMessage</td>
<td>OutboundSMSFlashMessage</td>
<td>Choice</td>
<td>Included if a Flash SMS 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.</td>
</tr>
</tbody>
</table>
This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate outbound SMS message request creation in such situations.

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

<table>
<thead>
<tr>
<th>resourceURL</th>
<th>xsd:anyURI</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>common:Link[0..unbounded]</td>
<td>Yes</td>
</tr>
<tr>
<td>deliveryInfoList</td>
<td>DeliveryInfoList</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST be also included in responses to any HTTP method that returns an entity body, and in PUT requests.

Links to other resources that are in relationship with the resource.

The delivery information (filled in by the server).

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

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 (POL1019: Policy error SHALL be used in case the message is rejected, see section 7.2).

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

Regarding the clientCorrelator field, the note in section 5.2.2.5 applies.

### 5.2.2.9 Type: OutboundSMSTextMessage

Content of an outbound textual SMS message.

<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>
5.2.2.10 **Type: OutboundSMSSBinaryMessage**

Content of an outbound binary SMS message.

<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>

5.2.2.11 **Type: OutboundSMSLogoMessage**

Content of an outbound SMS Logo message.

<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>

5.2.2.12 **Type: OutboundSMSRingToneMessage**

Content of an outbound SMS Ringtone message.

<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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: In the RTX Ringtone Specification, an RTX file is a text file, containing the ringtone 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>

5.2.2.13 **Type: OutboundSMSFlashMessage**

Content of an outbound Flash SMS message.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flashMessage</td>
<td>xsd:string</td>
<td>No</td>
<td>Content of Flash message</td>
</tr>
</tbody>
</table>

5.2.2.14 **Type: DeliveryInfoList**

List of delivery information records for an outbound SMS request.
### 5.2.2.15 Type: DeliveryInfoNotification

Notification about changes in the delivery information of an outbound SMS request.

<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>The <code>callbackData</code> element if it was passed by the application in the 'receiptRequest' element when creating an outbound SMS message request. See [REST_NetAPI_Common].</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. For example we can have a link to the original outbound message request.</td>
</tr>
</tbody>
</table>

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

### 5.2.2.16 Type: DeliveryInfo

Delivery information of an outbound SMS request regarding one recipient address.

<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 (e.g. 'sip' URI, 'tel' URI, 'acr' URI)</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>
5.2.2.17 Type: DeliveryReceiptSubscriptionList

List of subscriptions to notifications about changes in the delivery information of an outbound SMS request.

<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 response bodies.

5.2.2.18 Type: DeliveryReceiptSubscription

Individual subscription to notifications about changes in the delivery information of an outbound SMS request.

<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 endpoint and parameters</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 can use to tag this particular resource representation during a request to create a resource on the server. This element MAY be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.</td>
</tr>
</tbody>
</table>
A root element named deliveryReceiptSubscription of type DeliveryReceiptSubscription is allowed in request and/or response bodies.

Regarding the clientCorrelator field, the note in section 5.2.2.5 applies.

### 5.2.3 Enumerations

#### 5.2.3.1 Enumeration: DeliveryStatus

Delivery status enumeration.

<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</td>
</tr>
<tr>
<td></td>
<td>transition to one of the preceding states.</td>
</tr>
<tr>
<td>DeliveredToNetwork</td>
<td>Successful delivery to the network entity responsible for distributing the</td>
</tr>
<tr>
<td></td>
<td>short message further in the network</td>
</tr>
<tr>
<td>DeliveryNotificationNotSupported</td>
<td>Unable to provide delivery receipt notification. NotifyMessageDeliveryReceipt</td>
</tr>
<tr>
<td></td>
<td>function will provide DeliveryNotificationNotSupported to indicate that</td>
</tr>
<tr>
<td></td>
<td>delivery receipt for the specified address in a send message request is not</td>
</tr>
<tr>
<td></td>
<td>supported.</td>
</tr>
</tbody>
</table>

#### 5.2.3.2 Enumeration: SmsFormat

SMS format enumeration.

<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>
5.2.3.3 Enumeration: RetrievalOrder

Retrieval order enumeration.

<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>

5.2.4 Values of the Link “rel” attribute

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

- 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

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

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

5.3.1 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
  
  \[\text{http://\{serverRoot\}/smsmessaging/\{apiVersion\}/outbound/\{senderAddress\}/requests}\]

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

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

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

The resources:
To start subscription to notifications for inbound SMS messages, create new resource under
http://{serverRoot}/smsmessaging/{apiVersion}/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}/smsmessaging/{apiVersion}/inbound/subscriptions/{subscriptionId}

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. Alternatively, the application obtains the notifications using a Notification Channel [REST_NetAPI_NotificationChannel].
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
  
  http://{serverRoot}/smsmessaging/{apiVersion}/inbound/registrations/{registrationId}/messages

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

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.
6. Detailed specification of the resources

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

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

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

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

- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it MUST be defined according to [IETF_ACR_draft], i.e. it MUST include the protocol prefix ‘acr:’ followed by the ACR.
  - The ACR ‘auth’ is a supported reserved keyword, and MUST NOT be assigned as an ACR to any particular end user. See G.1.2 for details regarding the use of this reserved keyword.

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

6.1 Resource: Inbound SMS message requests for a given registration

The resource used is:

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

This resource is used for checking for incoming messages using retrieval critera that are setup in advance during provisioning process for a particular client.

6.1.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>registrationId</td>
<td>Reference to the off-line retrieval criteria provisioned in advance and known to the client application</td>
</tr>
</tbody>
</table>

See section 6 for a statement on the escaping of reserved characters in URL variables.
6.1.2  Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.1.3  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 method for each individual message (see DELETE on inbound SMS message).

Supported parameters in the query string of the Request URL 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 if there are more then one pending.</td>
</tr>
</tbody>
</table>

6.1.3.1  Example 1: Inbound message delivery  (Informative)

6.1.3.1.1  Request

GET /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=2 HTTP/1.1
Accept: application/xml
Host: example.com

6.1.3.1.2  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:inboundSMSMessageList xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <inboundSMSMessage>
    <destinationAddress>tel:+19585550120</destinationAddress>
    <senderAddress>tel:+19585550121</senderAddress>
    <message>First simple message</message>
    <dateTime>2009-11-19T12:00:00</dateTime>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001</resourceURL>
    <messageId>msg001</messageId>
  </inboundSMSMessage>
  <inboundSMSMessage>
    <destinationAddress>tel:+19585550122</destinationAddress>
    <senderAddress>tel:+19585550123</senderAddress>
    <message>Second simple message</message>
    <dateTime>2009-11-19T12:00:00</dateTime>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg002</resourceURL>
    <messageId>msg002</messageId>
  </inboundSMSMessage>
</sms:inboundSMSMessageList>
6.1.3.2  Example 2: maxBatchSize exceeding the allowed size  (Informative)

6.1.3.2.1  Request

GET /exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=5000 HTTP/1.1
Accept: application/xml
Host: example.com

6.1.3.2.2  Response

HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <link rel="InboundSMSMessageList" href="http://example.com/exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=50000"/>
  <policyException>
    <messageId>POL1020</messageId>
    <text>MaxBatchSize exceeded. The maximum allowed maxBatchSize is %1.</text>
    <variables>20</variables>
  </policyException>
</common:requestError>

6.1.4  PUT

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

6.1.5  POST

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

6.1.6  DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC2616].
6.2 Resource: Inbound SMS messages retrieve and delete using registration

The resource used is:

http://{serverRoot}/smmsmessaging/{apiVersion}/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages

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

6.2.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL.</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>registrationId</td>
<td>Reference to the off-line retrieval criteria provisioned in advance and known to the client application</td>
</tr>
</tbody>
</table>

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

6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.2.3 GET

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

6.2.4 PUT

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

6.2.5 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.

6.2.5.1 Example: Retrieve and delete using registration (Informative)

6.2.5.1.1 Request

```
POST /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/retrieveAndDeleteMessages HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:inboundSMSMessageRetrieveAndDeleteRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
    <retrievalOrder>OldestFirst</retrievalOrder>
    <maxBatchSize>3</maxBatchSize>
</sms:inboundSMSMessageRetrieveAndDeleteRequest>
```

6.2.5.1.2 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:netapi:sms:1">
    <!-- SMS -->
    <inboundSMSMessage>
        <destinationAddress>tel:+19585550120</destinationAddress>
        <senderAddress>tel:+19585550121</senderAddress>
        <message>First simple message</message>
        <messageId>msg001</messageId>
        <!-- no message resourceURL because SMS will be deleted from server immediately after operation is completed -->
    </inboundSMSMessage>
    <!-- SMS -->
    <inboundSMSMessage>
        <destinationAddress>tel:+19585550122</destinationAddress>
        <senderAddress>tel:+19585550123</senderAddress>
        <message>Second simple message</message>
        <messageId>msg002</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://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/retrieveAndDeleteMessages</resourceURL>
</sms:inboundSMSMessageList>
```
6.2.6 DELETE

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

6.3 Resource: Inbound SMS message for a given registration

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

6.3.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL.</td>
</tr>
<tr>
<td></td>
<td>Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined</td>
</tr>
<tr>
<td></td>
<td>in section 5.1.</td>
</tr>
<tr>
<td>registrationId</td>
<td>Reference to the off-line retrieval criteria provisioned in advance and known</td>
</tr>
<tr>
<td></td>
<td>to the client application</td>
</tr>
<tr>
<td>messageId</td>
<td>Unique message identifier generated by server</td>
</tr>
</tbody>
</table>

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

6.3.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.3.3 GET

Read one message from gateway storage. Message is not deleted. DELETE method needs to be executed to confirm delivery and free resources occupied by the message.

6.3.3.1 Example 1: Inbound messages for a given registration (Informative)

6.3.3.1.1 Request

This example shows also an alternative way to indicate desired content type in response from the server, by using URL query parameter “?resFormat” which is described in [REST_NetAPI_Common].

GET /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com
### 6.3.3.1.2 Response

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

```xml  
<?xml version="1.0" encoding="UTF-8"?>  
<sms:inboundSMSMessage xmlns:sms="urn:oma:xml:rest:netapi:sms:1">  
  <destinationAddress>tel:+19585550120</destinationAddress>  
  <senderAddress>tel:+19585550121</senderAddress>  
  <message>First simple message</message>  
  <dateTime>2009-11-19T12:00:00</dateTime>  
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001</resourceURL>  
  <messageId>msg001</messageId>  
</sms:inboundSMSMessage>  
```

### 6.3.3.2 Example 2: Invalid (non-existing) messageId  
(Informative)

#### 6.3.3.2.1 Request

GET /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001 HTTP/1.1  
Accept: application/xml  
Host: example.com  

#### 6.3.3.2.2 Response

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

```xml  
<?xml version="1.0" encoding="UTF-8"?>  
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">  
  <link rel="InboundSMSMessage" href="http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001" />  
  <serviceException>  
    <messageId>SVC0004</messageId>  
    <text>No valid addresses provided in message part %1</text>  
    <variables>msg001</variables>  
  </serviceException>  
</common:requestError>  
```

### 6.3.4 PUT

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

6.3.6 DELETE

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

6.3.6.1 Example: Remove message from gateway storage (Informative)

6.3.6.1.1 Request

DELETE /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001 HTTP/1.1
Accept: application/xml
Host: example.com

6.3.6.1.2 Response

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

6.4 Resource: Inbound SMS message subscriptions

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

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

This resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST_NetAPI_NotificationChannel]) before creating a subscription.

6.4.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL.</td>
</tr>
<tr>
<td></td>
<td>Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined</td>
</tr>
<tr>
<td></td>
<td>in section 5.1.</td>
</tr>
</tbody>
</table>

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

6.4.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.
6.4.3 GET

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

6.4.3.1 Example: Read active subscriptions

6.4.3.1.1 Request

GET /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com

6.4.3.1.2 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:subscriptionList xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <subscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <destinationAddress>tel:+19585550120</destinationAddress>
    <criteria>Urgent*</criteria>
    <clientCorrelator>67891</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001</resourceURL>
  </subscription>
  <subscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
      <callbackData>54321</callbackData>
    </callbackReference>
    <destinationAddress>tel:+19585550121</destinationAddress>
    <criteria>Urgent*</criteria>
    <clientCorrelator>67892</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub002</resourceURL>
  </subscription>
</sms:subscriptionList>

6.4.4 PUT

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

6.4.5 POST

This operation is used to create a new inbound message subscription for the particular client.
The notifyURL in the callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]).

**6.4.5.1 Example 1: Create inbound SMS message subscription, returning a representation of created resource** (Informative)

### 6.4.5.1.1 Request

```http
POST /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:subscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>tel:+19585550120</destinationAddress>
  <criteria>Urgent*</criteria>
  <clientCorrelator>67893</clientCorrelator>
</sms:subscription>
```

### 6.4.5.1.2 Response

```http
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001
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:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>tel:+19585550120</destinationAddress>
  <criteria>Urgent*</criteria>
  <clientCorrelator>67893</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001</resourceURL>
</sms:subscription>
```
6.4.5.2 Example 2: Create inbound SMS message subscription, returning the location of created resource (Informative)

6.4.5.2.1 Request

POST /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com


6.4.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT


6.4.6 DELETE

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

6.5 Resource: Individual inbound SMS message subscription

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

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

6.5.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>subscriptionId</td>
<td>Identifier of the subscription</td>
</tr>
</tbody>
</table>

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

### 6.5.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

### 6.5.3 GET

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

#### 6.5.3.1 Example: Read individual subscription (Informative)

**6.5.3.1.1 Request**

```
GET /exampleAPI/smssmessaging/v1/inbound/subscriptions/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
```

**6.5.3.1.2 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:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>tel:+19585550120</destinationAddress>
  <criteria>Urgent*</criteria>
  <clientCorrelator>67893</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/smssmessaging/v1/inbound/subscription/sub001</resourceURL>
</sms:subscription>
```

### 6.5.4 PUT

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

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

6.5.6 DELETE

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

6.5.6.1 Example: Delete subscription (Informative)

6.5.6.1.1 Request

```
DELETE /exampleAPI/smsmessaging/v1/inbound/subscriptions/sub000 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.5.6.1.2 Response

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

6.6 Resource: Client notification about inbound SMS message

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

Note: In the case when the client has set up a Notification Channel to obtain the notifications, the client needs to use the mechanisms described in [REST_NetAPI_NotificationChannel], instead of the mechanism described below in section 6.6.5.

6.6.1 Request URL variables

Client provided.

6.6.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

6.6.3 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 [RFC2616].

6.6.4 PUT

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

This operation is used to notify client about message arrival.

6.6.5.1  Example: Notify client about message arrival  (Informative)

6.6.5.1.1  Request

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

<?xml version="1.0" encoding="UTF-8"?>
<sms:inboundSMSMessageNotification xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackData>12345</callbackData>
  <inboundSMSMessage>
    <destinationAddress>tel:+19585550120</destinationAddress>
    <senderAddress>tel:+19585550121</senderAddress>
    <message>First simple message</message>
    <dateTime>2009-11-19T12:00:00</dateTime>
    <messageId>msg001</messageId>
  </inboundSMSMessage>
</sms:inboundSMSMessageNotification>

6.6.5.1.2  Response

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

6.6.6  DELETE

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

6.7  Resource: Outbound SMS message requests

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

This resource is used for sending outbound messages.

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

6.7.1  Request URL variables

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

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

© 2013 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
6.7.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.7.3 GET

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

6.7.3.1 Example: Retrieve list of pending outbound messages (Informative)

6.7.3.1.1 Request

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Host: example.com

6.7.3.1.2 Response

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequestList xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <outboundSMSMessageRequest>
    <address>tel:+19585550101</address>
    <senderAddress>tel:+19585550151</senderAddress>
    <outboundSMSTextMessage>
      <message>Let's have a REST.</message>
    </outboundSMSTextMessage>
    <clientCorrelator>67891</clientCorrelator>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req001</resourceURL>
    <deliveryInfoList>
      <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req001/deliveryInfos</resourceURL>
      <deliveryInfo>
        <address>tel:+19585550101</address>
        <deliveryStatus>DeliveredToNetwork</deliveryStatus>
      </deliveryInfo>
    </deliveryInfoList>
  </outboundSMSMessageRequest>
</sms:outboundSMSMessageRequestList>
<outboundSMSMessageRequest>
  <address>tel:+19585550102</address>
  <address>tel:+19585550103</address>
  <senderAddress>tel:+19585550151</senderAddress>
  <outboundSMSTextMessage>
    <message>Let's have a REST.</message>
  </outboundSMSTextMessage>
  <clientCorrelator>67892</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req002</resourceURL>
  <deliveryInfoList>
    <deliveryInfo>
      <address>tel:+19585550102</address>
      <deliveryStatus>DeliveredToTerminal</deliveryStatus>
    </deliveryInfo>
    <deliveryInfo>
      <address>tel:+19585550103</address>
      <deliveryStatus>DeliveredToNetwork</deliveryStatus>
    </deliveryInfo>
  </deliveryInfoList>
</outboundSMSMessageRequest>

6.7.4 PUT

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

6.7.5 POST

This operation is used to create outgoing message request.

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

6.7.5.1 Example 1: Create outbound message request, returning representation of created resource in response (Informative)

6.7.5.1.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
</sms:outboundSMSMessageRequest>
<senderAddress>tel:+19585550151</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>

6.7.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>tel:+19585550151</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://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000/deliveryInfos</resourceURL>
</sms:outboundSMSMessageRequest>

6.7.5.2 Example 2: Create outbound message request, returning location of created resource in response (Informative)

6.7.5.2.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
    <address>tel:+19585550101</address>
    <address>tel:+19585550104</address>
    <senderAddress>tel:+19585550151</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>

6.7.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000</resourceURL>
</common:resourceReference>

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

6.7.5.3.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
    <address>tel:+19585550101</address>
    <address>tel:+19585550104</address>
    <senderAddress>tel:+19585550151</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>
<notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
</receiptRequest>
<outboundSMSTextMessage>
  <message>Example Text Message</message>
</outboundSMSTextMessage>
<clientCorrelator>67893</clientCorrelator>
</sms:outboundSMSMessageRequest>

6.7.5.2 Response

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

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

6.7.5.4 Example 4: Multiple addresses partial success, with deliveryInfoList in response (Informative)

6.7.5.4.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>tel:+19585550151</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>
6.7.5.4.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smmsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>tel:+19585550151</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://example.com/exampleAPI/smmsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000</resourceURL>
  <deliveryInfoList>
    <deliveryInfo>
      <address>tel:+19585550101</address>
      <deliveryStatus>MessageWaiting</deliveryStatus>
    </deliveryInfo>
    <deliveryInfo>
      <address>tel:+19585550104</address>
      <deliveryStatus>DeliveryImpossible</deliveryStatus>
    </deliveryInfo>
  </deliveryInfoList>
</sms:outboundSMSMessageRequest>

6.7.5.5 Example 5: Multiple addresses partial success, without deliveryInfoList in response

6.7.5.5.1 Request

POST /exampleAPI/smmsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>tel:+19585550151</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://example.com/exampleAPI/smmsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000</resourceURL>
  <deliveryInfo>
    <address>tel:+19585550101</address>
    <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
  <deliveryInfo>
    <address>tel:+19585550104</address>
    <deliveryStatus>DeliveryImpossible</deliveryStatus>
  </deliveryInfo>
</sms:outboundSMSMessageRequest>
6.7.5.5.2 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://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>tel:+19585550151</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://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000</resourceURL>
</sms:outboundSMSMessageRequest>

6.7.5.6 Example 6: Create outbound message using SHORT CODE as senderAddress, returning a representation of created resource (Informative)

6.7.5.6.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/72654/requests HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
</sms:outboundSMSMessageRequest>
<senderAddress>72654</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>

6.7.5.6.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/72654/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <address>tel:+19585550101</address>
  <address>tel:+19585550104</address>
  <senderAddress>72654</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://example.com/exampleAPI/smsmessaging/v1/outbound/72654/requests/req000</resourceURL>
  <deliveryInfoList>
    <deliveryInfo>
      <address>tel:+19585550101</address>
      <deliveryStatus>MessageWaiting</deliveryStatus>
    </deliveryInfo>
    <deliveryInfo>
      <address>tel:+19585550104</address>
      <deliveryStatus>MessageWaiting</deliveryStatus>
    </deliveryInfo>
  </deliveryInfoList>
</sms:outboundSMSMessageRequest>

6.7.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC2616].
6.8 Resource: Outbound SMS message request and delivery status

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

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

6.8.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>Sender identifier. Examples: 72654 (SHORT CODE [REST_NetAPI_Common]), tel:+19585550100, acr:pseudonym123</td>
</tr>
<tr>
<td>requestId</td>
<td>Outbound message request Id generated by server</td>
</tr>
</tbody>
</table>

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

6.8.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.8.3 GET

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

6.8.3.1 Example: Get message delivery status (Informative)

6.8.3.1.1 Request

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000 HTTP/1.1
Accept: application/xml
Host: example.com

6.8.3.1.2 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:outboundSMSMessageRequest xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
6.8.4 PUT

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

6.8.5 POST

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

6.8.6 DELETE

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

6.9 Resource: Outbound SMS message delivery status

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

This resource is used to request outbound message delivery status.

6.9.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>Sender identifier. Examples: 72654 (SHORT CODE [REST_NetAPI_Common]), tel:+19585550100, acr:pseudonym123</td>
</tr>
<tr>
<td>requestId</td>
<td>Outbound message request Id generated by server.</td>
</tr>
</tbody>
</table>

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

### 6.9.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

### 6.9.3 GET

This operation is used to retrieve outgoing message delivery status.

#### 6.9.3.1 Example: Get message delivery status (Informative)

**6.9.3.1.1 Request**

```
GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000/deliveryInfos HTTP/1.1
Accept: application/xml
Host: example.com
```

**6.9.3.1.2 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:deliveryInfoList xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000/deliveryInfos</resourceURL>
  <deliveryInfo>
    <address>tel:+19585550101</address>
    <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
  <deliveryInfo>
    <address>tel:+19585550104</address>
    <deliveryStatus>MessageWaiting</deliveryStatus>
  </deliveryInfo>
</sms:deliveryInfoList>
```
6.9.4 PUT

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

6.9.5 POST

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

6.9.6 DELETE

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

6.10 Resource: Outbound SMS message delivery notification subscriptions

The resource used is:

http://{serverRoot}/smsmessaging/{apiVersion}/outbound/{senderAddress}/subscriptions

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

This resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST NetAPI NotificationChannel]) before creating a subscription.

6.10.1 Request URL variables

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>Sender identifier. Examples: 72654 (SHORT CODE [REST NetAPI Common]), tel:+19585550100, acr:pseudonym123</td>
</tr>
</tbody>
</table>

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

6.10.2 Response Codes and Error Handling

For HTTP response codes, see [REST NetAPI Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.10.3 GET

This operation is used to read all outbound SMS delivery notification subscriptions for the particular client.
6.10.3.1 Example: Read delivery notification subscriptions (Informative)

6.10.3.1.1 Request

GET /exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com

6.10.3.1.2 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:deliveryReceiptSubscriptionList xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/</resourceURL>
  <deliveryReceiptSubscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
      <callbackData>12345</callbackData>
    </callbackReference>
    <filterCriteria>0102</filterCriteria>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000</resourceURL>
  </deliveryReceiptSubscription>
  <deliveryReceiptSubscription>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
      <callbackData>54321</callbackData>
    </callbackReference>
    <filterCriteria>0103</filterCriteria>
    <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000</resourceURL>
  </deliveryReceiptSubscription>
</sms:deliveryReceiptSubscriptionList>

6.10.4 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 [RFC2616].

6.10.5 POST

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

The notifyURL in the callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST_NetAPI_NotificationChannel]).
6.10.5.1 Example: Create outbound delivery notification subscription, using tel URI (Informative)

6.10.5.1.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com


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

6.10.5.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT


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 6.4.5.2.2).

6.10.5.2 Example: Create outbound delivery notification subscription, using ACR (Informative)

6.10.5.2.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/acr%3Apseudonym123/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/xml
Content-Length: nnnn
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/66666</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
</sms:deliveryReceiptSubscription>

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

6.10.5.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smssmessaging/v1/outbound/acr%3Apseudonym123/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/66666</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://example.com/exampleAPI/smssmessaging/v1/outbound/acr%3Apseudonym123/subscriptions/sub000</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 6.4.5.2.2).

6.10.6 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 [RFC2616].

6.11 Resource: Individual outbound SMS message delivery notification subscription

The resource used is:
http://[serverRoot]/smssmessaging/{apiVersion}/outbound/{senderAddress}/subscriptions/{subscriptionId}

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

The following request URL 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. Port and base path are OPTIONAL. Example: example.com/exampleAPI</td>
</tr>
<tr>
<td>apiVersion</td>
<td>Version of the API client wants to use. The value of this variable is defined in section 5.1.</td>
</tr>
<tr>
<td>senderAddress</td>
<td>Sender identifier. Examples: 72654 (SHORT CODE [REST_NetAPI_Common]), tel:+19585550100, acr:pseudonym123</td>
</tr>
<tr>
<td>subscriptionId</td>
<td>Identifier of the subscription</td>
</tr>
</tbody>
</table>

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

6.11.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Short Messaging, see section 7.

6.11.3 GET

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

6.11.3.1 Example: Read delivery notification subscription (Informative)

6.11.3.1.1 Request

GET /exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000 HTTP/1.1
Accept: application/xml
Host: example.com

6.11.3.1.2 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:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://example.com/exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000</resourceURL>
</sms:deliveryReceiptSubscription>
6.11.4 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 [RFC2616].

6.11.5 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 [RFC2616].

6.11.6 DELETE

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

6.11.6.1 Example: Delete subscription for a client (Informative)

6.11.6.1.1 Request

```
DELETE /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.11.6.1.2 Response

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

6.12 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. The RESTful ShortMessaging API does not make any assumption about the structure of this URL. If this URL is a Client-side Notification URL, the server will POST notifications directly to it. If this URL is a Server-side Notification URL, the server uses it to determine the address of the Notification Server to which the notifications will subsequently be POSTed. The way the server determines the address of the Notification Server is out of scope of this specification.

Note: In the case when the client has set up a Notification Channel to obtain the notifications, the client needs to use the mechanisms described in [REST_NetAPI_NotificationChannel], instead of the mechanism described below in section 6.12.5.

6.12.1 Request URL variables

Client provided.

6.12.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].
6.12.3 GET

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

6.12.4 PUT

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

6.12.5 POST

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

6.12.5.1 Example: Notify client about message delivery status (Informative)

6.12.5.1.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
  <callbackData>12345</callbackData>
  <deliveryInfo>
    <address>tel:+19585550101</address>
    <deliveryStatus>DeliveredToNetwork</deliveryStatus>
  </deliveryInfo>
  <link rel="DeliveryReceiptSubscription"
    href="http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000"/>
</sms:deliveryInfoNotification>
```

6.12.5.1.2 Response

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

6.12.6 DELETE

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

7.1 Service Exceptions

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

7.1.1 SVC0280: Message too long

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>SVC0280</td>
</tr>
<tr>
<td>Text</td>
<td>Message too long. Maximum length is %1 characters</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Number of characters allowed in a message</td>
</tr>
<tr>
<td>HTTP status code(s)</td>
<td>403 Forbidden</td>
</tr>
</tbody>
</table>

7.1.2 SVC0281: Unrecognized data format

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>SVC0281</td>
</tr>
<tr>
<td>Text</td>
<td>Data format not recognized for message part %1</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Message part with the unrecognized data</td>
</tr>
<tr>
<td>HTTP status code(s)</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

7.1.3 SVC0283: Delivery Receipt Notification not supported

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>SVC0283</td>
</tr>
<tr>
<td>Text</td>
<td>Delivery Receipt Notification not supported</td>
</tr>
<tr>
<td>Variables</td>
<td></td>
</tr>
<tr>
<td>HTTP status code(s)</td>
<td>403 Forbidden</td>
</tr>
</tbody>
</table>

7.2 Policy Exceptions

For common Policy Exceptions refer to [REST_NetAPI_Common].

The following additional Policy Exception codes are defined for the RESTful Short Messaging API.
### 7.2.1 POL1019: Binary SMS not allowed

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>POL1019</td>
</tr>
<tr>
<td>Text</td>
<td>Binary SMS is not allowed.</td>
</tr>
<tr>
<td>Variables</td>
<td>None</td>
</tr>
<tr>
<td>HTTP status code(s)</td>
<td>403 Forbidden</td>
</tr>
</tbody>
</table>

### 7.2.2 POL1020: MaxBatchSize exceeded

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>POL1020</td>
</tr>
<tr>
<td>Text</td>
<td>MaxBatchSize exceeded. The maximum allowed maxBatchSize is %1.</td>
</tr>
<tr>
<td>Variables</td>
<td>%1 Allowed maximum value for maxBatchSize</td>
</tr>
<tr>
<td>HTTP status code(s)</td>
<td>403 Forbidden</td>
</tr>
</tbody>
</table>
## Appendix A. Change History

### A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>n/a</td>
<td>No prior version</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>
<tbody>
<tr>
<td>Draft Versions: OMA-TS-REST_NetAPI_ShortMessaging-V1_0</td>
<td>03 May 2011</td>
<td>Many</td>
<td>Structural changes to fit the OMA RESTful Network API release. This version inherits the technical content of OMA-TS-ParlayREST_ShortMessaging-V1_1-20110111-C and applies changes according to ARC INP 30R01, 98R02, 155R01, 156R01, 71R01, 175R01, 186, 187R02 and 159R03</td>
</tr>
<tr>
<td></td>
<td>08 Dec 2011</td>
<td>5.1</td>
<td>Editorial change. Fixed a typo in the sentence above the resource tables; &quot;table gives&quot; replaced with &quot;tables give&quot;</td>
</tr>
<tr>
<td>Candidate Version: OMA-TS-REST_NetAPI_ShortMessaging-V1_0</td>
<td>20 Dec 2011</td>
<td>n/a</td>
<td>Status changed to Candidate by TP TP ref # OMA-TP-2011-0444-INP_REST_NetAPI_ShortMessaging_1_0_ERP_and_ETR_for_Candidate_approval</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>
</tbody>
</table>
| Draft Versions: OMA-TS-REST_NetAPI_ShortMessaging-V1_0 | 18 Jul 2012 | 5, 5.3, 5.3.2, 6.1.1, 6.1.2, 6.2.1, 6.2.2, 6.3.1, 6.3.2, 6.4.1, 6.4.2, 6.5.1, 6.5.2, 6.7.1, 6.7.2, 6.8, 6.8.1, 6.8.2, 6.9.1, 6.9.2, 6.10.1, 6.10.2, 6.11.1, 6.11.2, 7, G.1.1.3 | Status changed to Draft  
Incorporated CRs:  
OMA-ARC-REST-NetAPI-2012-0165-CR_SMS_POST_or_NOTIFY_changes  
OMA-ARC-REST-NetAPI-2012-0166-CR_SMS_TS_Adding_Section_7_A053  
Editorial changes |
|                     |            | 24 Aug 2012 | 5.2.2.5, 5.2.2.8, 5.2.2.17, C.1, C.2, C.3 | Incorporated CR:  
OMA-ARC-REST-NetAPI-2012-0232-CR_SMS_TS_issue_20_clientCorrelator_resolution  
Editorial changes |
|                     |             | 02 Oct 2012 | 5.2.2.8, 5.2.2.10, 5.2.2.13, 6.1.3.2.2, 6.3.3.2.2, 6.7.5.3.2, 7.2, 7.2.1, 7.2.2, D.2, D.5, D.16 | Incorporated CRs:  
OMA-ARC-REST-NetAPI-2012-0212-CR_Followup_for_INP_200_TS_SMS  
OMA-ARC-REST-NetAPI-2012-0257R01-CR_ShortMessaging_support_for_Flash_SMS  
Editorial changes |
|                     |             | 17 Oct 2012 | 7.1.2 | Incorporated CR:  
OMA-ARC-REST-NetAPI-2012-0272-CR_TS_SMS_HTTP_response_code_fix  
Editorial changes |
|                     |             | 13 Dec 2012 | 3.2, 4.1, 5.1, 5.2.2, 5.2.3, 5.3.2, 6, 6.1.1, 6.1.3, 6.2.1, 6.3.1, 6.4.1, 6.5.1, 6.6, 6.7.1, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12, 7.2, B, C.1, C.2, C.3, E, G.1.1.1, G.1.1.3, G.1.2 | Incorporated CR:  
OMA-ARC-REST-NetAPI-2012-0289R01-CR_SMS_TS_blueprint_for_longpolling_and_authorization  
Template changed to OMA-TEMPLATE-TS_RESTful_Network_API-20120813-I  
Editorial changes |
|                     |            | 08 Feb 2013 | 2.2 | Reference to OMA Dictionary updated to version 2.9.  
Template updated. |
<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>Date</th>
<th>Sections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Version</td>
<td>19 Feb 2013</td>
<td>n/a</td>
<td>Status changed to Candidate by TP</td>
</tr>
<tr>
<td>OMA-TS-REST_NetAPI_ShortMessaging-V1_0</td>
<td></td>
<td></td>
<td>TP Ref # OMA-TP-2013-0041-INP_REST_NetAPI_ShortMessaging_V1_0_ERP_for_Candidate_re_approval</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for REST.SMS Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-SUPPORT-S-001-M</td>
<td>Support for the RESTful SMS API</td>
<td>5, 6</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-SUPPORT-S-002-M</td>
<td>Support for the XML request &amp; response format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REST-SMS-SUPPORT-S-003-M</td>
<td>Support for the JSON request &amp; response format</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-SUPPORT-S-004-O</td>
<td>Support for the application/x-www-form-urlencoded format</td>
<td>Appendix C</td>
<td></td>
</tr>
</tbody>
</table>

B.1.1 SCR for REST.SMS.Inbound.Registration Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-INB-OFF-S-001-M</td>
<td>Support for reliable inbound message delivery</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-INB-OFF-S-002-M</td>
<td>Retrieve messages from server - GET</td>
<td>6.1.3</td>
<td></td>
</tr>
</tbody>
</table>

B.1.2 SCR for REST.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>REST-SMS-INB-OFF-RETDEL-S-002-O</td>
<td>Retrieve messages from server - POST</td>
<td>6.2.5</td>
<td></td>
</tr>
</tbody>
</table>

B.1.3 SCR for REST.SMS.Individual.Inbound Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-IND-INB-S-001-M</td>
<td>Support for inbound individual message delivery</td>
<td>6.3</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-IND-INB-S-002-O</td>
<td>Retrieve one message from server - GET</td>
<td>6.3.3</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-IND-INB-S-003-M</td>
<td>Confirm and delete retrieved message from server - DELETE</td>
<td>6.3.6</td>
<td></td>
</tr>
</tbody>
</table>
### B.1.4 SCR for REST.SMS.Inbound.Subscr Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-INB-ONL-SUBSCR-S-001-M</td>
<td>Support inbound subscriptions</td>
<td>6.4</td>
</tr>
<tr>
<td>REST-SMS-INB-ONL-SUBSCR-S-002-O</td>
<td>Read active subscriptions - GET</td>
<td>6.4.3</td>
</tr>
<tr>
<td>REST-SMS-INB-ONL-SUBSCR-S-003-M</td>
<td>Create inbound message subscription - POST (XML or JSON)</td>
<td>6.4.5</td>
</tr>
<tr>
<td>REST-SMS-INB-ONL-SUBSCR-S-004-O</td>
<td>Create inbound message subscription – POST (application/x-www-form-urlencoded)</td>
<td>C.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-INB-INDON-SUBSCR-S-001-M</td>
<td>Support for control and read access to individual inbound subscription</td>
<td>6.5</td>
</tr>
<tr>
<td>REST-SMS-INB-INDON-SUBSCR-S-002-O</td>
<td>Read individual inbound subscription - GET</td>
<td>6.5.3</td>
</tr>
<tr>
<td>REST-SMS-INB-INDON-SUBSCR-S-003-M</td>
<td>Update individual inbound subscriptions - DELETE</td>
<td>6.5.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-INB-NOTIF-S-001-M</td>
<td>Support for notifying application about inbound messages</td>
<td>6.6</td>
</tr>
<tr>
<td>REST-SMS-INB-NOTIF-S-002-M</td>
<td>Notify application about inbound message arrival - POST</td>
<td>6.6.5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-OUTB-S-001-M</td>
<td>Support for outbound SMS messages</td>
<td>6.7</td>
</tr>
<tr>
<td>REST-SMS-OUTB-S-002-O</td>
<td>Retrieve list of pending outgoing message requests - GET</td>
<td>6.7.3</td>
</tr>
<tr>
<td>REST-SMS-OUTB-S-003-M</td>
<td>Create outgoing message request</td>
<td>6.7.5</td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>POST (XML and JSON)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REST-SMS-OUTB-S-004-O</td>
<td>Create outgoing message request - POST (application/x-www-form-urlencoded)</td>
<td>C.1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-OUTB-MSGDELSTAT-S-001-O</td>
<td>Support for requesting an outbound SMS message and its delivery status</td>
<td>6.8</td>
<td>REST-SMS-OUTB-MSGDELSTAT-S-002-O</td>
</tr>
<tr>
<td>REST-SMS-OUTB-MSGDELSTAT-S-002-O</td>
<td>Retrieve outgoing message delivery status - GET</td>
<td>6.8.3</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-OUTB-DELSTAT-S-001-M</td>
<td>Support for requesting delivery status of outbound SMS messages</td>
<td>6.9</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-OUTB-DELSTAT-S-002-M</td>
<td>Retrieve outgoing message delivery status - GET</td>
<td>6.9.3</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-OUTB-SUBSCR-S-001-M</td>
<td>Support for outbound subscriptions for a particular client</td>
<td>6.10</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-OUTB-SUBSCR-S-002-O</td>
<td>Read all outbound SMS delivery notification subscriptions - GET</td>
<td>6.10.3</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-OUTB-SUBSCR-S-003-M</td>
<td>Create new outbound message subscription – POST (XML and JSON)</td>
<td>6.10.5</td>
<td></td>
</tr>
<tr>
<td>REST-SMS-OUTB-SUBSCR-S-004-O</td>
<td>Create new outbound message subscription – POST (application/x-www-form-urlencoded)</td>
<td>C.2</td>
<td></td>
</tr>
</tbody>
</table>
### B.1.11 SCR for REST.SMS.Individual.Outbound.Subscr Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-IND-OUTB-IND-SUBSCR-S-001-M</td>
<td>Support for outbound subscriptions for a particular client</td>
<td>6.11</td>
</tr>
<tr>
<td>REST-SMS-IND-OUTB-IND-SUBSCR-S-002-O</td>
<td>Read individual SMS delivery notification subscription - GET</td>
<td>6.11.3</td>
</tr>
<tr>
<td>REST-SMS-IND-OUTB-IND-SUBSCR-S-003-M</td>
<td>Delete subscription for the client - DELETE</td>
<td>6.11.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST-SMS-OUTB-DELSTAT-NOTIF-S-001-M</td>
<td>Support for notifying application about delivery status of outbound messages</td>
<td>6.12</td>
</tr>
</tbody>
</table>
Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations

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

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

Names and values MUST follow the application/x-www-form-urlencoded character escaping rules at [W3C_URLENC].

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

- 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, see section 6.7.5.

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

The request parameters are as follows:

<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 (e.g. ’sip’ URI, ’tel’ URI, ’acr’ URI)</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 (e.g. ’sip’ URI, ’tel’ URI, ’acr’ URI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If senderAddress is also part of the request URL, the two MUST have the same value.</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>
<tr>
<td>notificationFormat</td>
<td>common:NotificationFormat</td>
<td>Yes</td>
<td>Default: XML Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This element SHOULD be present. Note: this allows the</td>
</tr>
</tbody>
</table>
client to recover from communication failures during resource creation and therefore avoids duplicate outbound SMS message request creation in such situations.

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

<table>
<thead>
<tr>
<th>Field</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 user's terminal as the originator of the message.</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: Create outbound message request, returning representation of created resource in response (Informative)

C.1.1.1 Request

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/xml
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Host: example.com

address=tel%3A%2B19585550101&
address=tel%3A%2B19585550104&
senderAddress=tel%3A%2B19585550151&
message= Example%20Text%20Message&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification&
notificationFormat=XML&
clientCorrelator=123456&
senderName=MyName

C.1.1.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: nnnn
C.2 Start delivery receipt notification

This REST method is used by the application to start the delivery receipt notifications, see section 6.10.5.

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

The 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>
<tr>
<td>notificationFormat</td>
<td>common:NotificationFormat</td>
<td>Yes</td>
<td>Default: XML. Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.</td>
</tr>
<tr>
<td>clientCorrelator</td>
<td>xsd:string</td>
<td>Yes</td>
<td>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</td>
</tr>
</tbody>
</table>

This element MAY be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations.

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

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

C.2.1 Example: Create outbound delivery notification subscription, using tel URI (Informative)

C.2.1.1 Request

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

filterCriteria=0102&
notifyURL=http://application.example.com/notifications/DeliveryInfoNotification/66666
```

C.2.1.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/66666</notifyURL>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000</resourceURL>
</sms:deliveryReceiptSubscription>
```

C.2.2 Example: Create outbound delivery notification subscription, using ACR (Informative)

C.2.2.1 Request

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

© 2013 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
C.2.2.2 Response

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/acr%3Apseudonym123/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<sms:deliveryReceiptSubscription xmlns:sms="urn:oma:xml:rest:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification/66666</notifyURL>
  </callbackReference>
  <filterCriteria>0102</filterCriteria>
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/outbound/acr%3Apseudonym123/subscriptions/sub000</resourceURL>
</sms:deliveryReceiptSubscription>

C.3 Start SMS notification

This REST method is used by the application to start the notification of received SMS, see section 6.4.5.

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

The 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</td>
<td>No</td>
<td>Destination address of SMS message (e.g. 'sip' URI, 'tel' URI, 'acr' URI)</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 can use to tag this particular resource representation during a request to create a resource on the server. This element MAY be present. Note: this allows the
client to recover from communication failures during resource creation and therefore avoids duplicate subscription creation in such situations.

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

This operation returns a result indicating whether the operation has been successful.

### C.3.1 Example: Create inbound SMS message subscription, returning a representation of created resource (Informative)

#### C.3.1.1 Request

Note that this example also illustrates the use of callbackData.

```
POST /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/xml
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn
Host: example.com

destinationAddress=tel%3A%2B19585550120&criteria=Vote&clientCorrelator=67893&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://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001
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:netapi:sms:1">
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/DeliveryInfoNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <destinationAddress>tel:+19585550120</destinationAddress>
  <criteria>Vote</criteria>
  <clientCorrelator>67893</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001</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 invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC4627].

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

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

D.1 Inbound message delivery (section 6.1.3.1)

Request:

```
GET /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=2 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

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

{"inboundSMSMessageList": {
   "inboundSMSMessage": [
      {
         "dateTime": "2009-11-19T12:00:00",
         "destinationAddress": "tel:+19585550120",
         "message": "First simple message",
         "messageId": "msg001",
         "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001",
         "senderAddress": "tel:+19585550121"
      },
      {
         "dateTime": "2009-11-19T12:00:00",
         "destinationAddress": "tel:+19585550122",
         "message": "Second simple message",
         "messageId": "msg002",
         "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg002",
         "senderAddress": "tel:+19585550123"
      }
   ],
   "numberOfMessagesInThisBatch": "2",
   "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages",
   "totalNumberOfPendingMessages": "20"
}}
```
D.2  maxBatchSize exceeding the allowed size (section 6.1.3.2)

Request:

GET /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=5000 HTTP/1.1
Accept: application/json
Host: example.com

Response:

HTTP/1.1 403 Forbidden
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"requestError": {
    "link": {
        "href": "http://example.com/exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages?maxBatchSize=5000",
        "rel": "InboundSMSMessageList"
    },
    "policyException": {
        "messageId": "POL1020",
        "text": "MaxBatchSize exceeded. The maximum allowed maxBatchSize is %1.",
        "variables": "20"
    }
}}

D.3  Retrieve and delete using registration (section 6.2.5.1)

Request:

POST /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/retrieveAndDeleteMessages HTTP/1.1
Accept: application/json
Content-Length: nnnn
Content-Type: application/json
Host: example.com

{"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": "tel:+19585550120",
        }
    ]
}}
D.4  Inbound messages for a given registration (section 6.3.3.1)

Request:

GET /exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages/msg001?resFormat=JSON HTTP/1.1
Host: example.com

Response:

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

{"inboundSMSMessage": {
    "dateTime": "2009-11-19T12:00:00",
    "destinationAddress": "tel:+19585550120",
    "message": "First simple message",
    "messageId": "msg001",
    "resourceURL": "http://example.com/exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages/msg001",
    "senderAddress": "tel:+19585550121"
}}

D.5  Invalid (non-existing) messageId (section 6.3.3.2)

Request:

GET /exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages/msg001 HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

{"nonExistingMessage": {
    "destinationAddress": "tel:+19585550122",
    "message": "Second simple message",
    "messageId": "msg002",
    "resourceURL": "http://example.com/exampleAPI/smssmessaging/v1/inbound/registrations/reg000/messages/retrieveAndDeleteMessages",
    "totalNumberOfPendingMessages": "200"
}}
D.6  Remove message from gateway storage (section 6.3.6.1)

Request:

DELETE /exampleAPI/smsmessaging/v1/inbound/registrations/reg000/messages/msg001 HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

D.7  Read active subscriptions (section 6.4.3.1)

Request:

GET /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

{"subscriptionList": {
  "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions",
  "subscription": [
    {
      "callbackReference": {
      ...
    }
  ]
}
D.8 Create inbound SMS message subscription, returning a representation of created resource (section 6.4.5.1)

Request:

POST /exampleAPI/smssmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"subscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
  "clientCorrelator": "67891",
  "criteria": "Urgent*",
  "destinationAddress": "tel:+19585550120",
  "resourceURL": "http://example.com/exampleAPI/smssmessaging/v1/inbound/subscriptions/sub001"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smssmessaging/v1/inbound/subscriptions/sub001
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"subscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
  "clientCorrelator": "67891",
  "criteria": "Urgent*",
  "destinationAddress": "tel:+19585550120",
  "resourceURL": "http://example.com/exampleAPI/smssmessaging/v1/inbound/subscriptions/sub001"
}}
D.9 Create inbound SMS message subscription, returning the location of created resource (section 6.4.5.2)

Request:

```json
POST /exampleAPI/smsmessaging/v1/inbound/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"subscription": {
    "callbackReference": {
        "callbackData": "12345",
        "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
    },
    "criteria": "Urgent*",
    "destinationAddress": "tel:+19585550120",
    "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/subscriptions/sub001"}}
```

D.10 Read individual subscription (section 6.5.3.1)

Request:

```http
GET /exampleAPI/smsmessaging/v1/inbound/subscriptions/sub000 HTTP/1.1
Accept: application/json
Host: example.com
```

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"
  },
  "clientCorrelator": "67893",
  "criteria": "Urgent",
  "destinationAddress": "tel:+19585550120",
  "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/inbound/subscription/sub001"}
}

D.11 Delete a subscription (section 6.5.6.1)

Request:
DELETE /exampleAPI/smsmessaging/v1/inbound/subscriptions/sub000 HTTP/1.1
Accept: application/json
Host: example.com

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

D.12 Notify client about message arrival (section 6.6.5.1)

Request:
POST /notifications/DeliveryInfoNotification HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: application.example.com

{"inboundSMSMessageNotification": {
  "callbackData": "12345",
  "inboundSMSMessage": {
    "dateTime": "2009-11-19T12:00:00",
    "destinationAddress": "tel:+19585550120",
    "message": "First simple message",
    "messageId": "msg001",
    "senderAddress": "tel:+19585550121"
  }
}}

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

D.13 Retrieve list of pending outbound messages (section 6.7.3.1)

Request:

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

{"outboundSMSMessageRequestList": {
  "outboundSMSMessageRequest": [
    {
      "address": "tel:+19585550101",
      "clientCorrelator": "67891",
      "deliveryInfoList": {
        "deliveryInfo": {
          "address": "tel:+19585550101",
          "deliveryStatus": "DeliveredToNetwork"
        },
        "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req001/deliveryInfos"
      },
      "outboundSMSTextMessage": {"message": "Let's have a REST."},
      "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req001",
      "senderAddress": "tel:+19585550151"
    },
    {
      "address": [
        "tel:+19585550102",
        "tel:+19585550103"
      ],
      "clientCorrelator": "67892",
      "deliveryInfoList": {
        "deliveryInfo": [
          {
            "address": "tel:+19585550102",
            "deliveryStatus": "DeliveredToTerminal"
          },
          {
            "address": "tel:+19585550103",
            "deliveryStatus": "DeliveredToNetwork"
          }
        ]
      }
    }
  ]
}
D.14 Create outbound message request, returning a representation of created resource in response (section 6.7.5.1)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "senderAddress": "tel:+19585550151",
  "senderName": "MyName"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "deliveryInfoList": [
  
  ]
}}
D.15 Create outbound message request, returning the location of created resource in response (section 6.7.5.2)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "senderAddress": "tel:+19585550151",
  "senderName": "MyName"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"resourceReference": {"resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000"}}
D.16 serviceException in case of single address or all multiple addresses failure (section 6.7.5.3)

Request:

```
POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {
    "message": "Example Text Message"},
  "receiptRequest": {
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "senderAddress": "tel:+19585550151",
  "senderName": "MyName"
}}
```

Response:

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

{"requestError": {
  "serviceException": {
    "messageId": "SVC0004",
    "text": "No valid addresses provided in message part %1",
    "variables": "address"
  }}
```

D.17 Multiple addresses partial success, with deliveryInfoList in response (section 6.7.5.4)

Request:

```
POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
}}
```
"clientCorrelator": "67893",
"outboundSMSTextMessage": {"message": "Example Text Message"},
"receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
"senderAddress": "tel:+19585550151",
"senderName": "MyName"
}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "deliveryInfoList": {
    "deliveryInfo": [
      {
        "address": "tel:+19585550101",
        "deliveryStatus": "MessageWaiting"
      },
      {
        "address": "tel:+19585550104",
        "deliveryStatus": "DeliveryImpossible"
      }
    ],
    "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000/deliveryInfos"
  },
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000",
  "senderAddress": "tel:+19585550151",
  "senderName": "MyName"
}}

D.18 Multiple addresses partial success, without deliveryInfoList in response (section 6.7.5.5)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com
D.19 Create outbound message using SHORT CODE as senderAddress, returning a representation of created resource (section 6.7.5.6)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/72654/requests HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"outboundSMSMessageRequest": {
  "address": [
    "tel:+19585550101",
    "tel:+19585550104"
  ],
  "clientCorrelator": "67893",
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification",
    "senderAddress": "tel:+19585550151",
    "senderName": "MyName"
  }
}}
"outboundSMSTextMessage": {"message": "Example Text Message"},
"receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
"senderAddress": "72654",
"senderName": "MyName"
}
}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/72654/requests/req000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"outboundSMSMessageRequest": {
  "address": ["tel:+19585550101", "tel:+19585550104"],
  "clientCorrelator": "67893",
  "deliveryInfoList": {
    "deliveryInfo": [
      {"address": "tel:+19585550101", "deliveryStatus": "MessageWaiting"},
      {"address": "tel:+19585550104", "deliveryStatus": "MessageWaiting"}
    ],
    "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/72654/requests/req000/deliveryInfos"
  },
  "outboundSMSTextMessage": {"message": "Example Text Message"},
  "receiptRequest": {"notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"},
  "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/72654/requests/req000",
  "senderAddress": "72654",
  "senderName": "MyName"
}}

D.20 Get message delivery status (section 6.8.3.1)

Request:

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000 HTTP/1.1
Accept: application/json
Host: example.com

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
D.21 Get message delivery status (section 6.9.3.1)

Request:

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/requests/req000/deliveryInfos HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

{"deliveryInfoList": {
  "deliveryInfo": [
  {
    "address": "tel:+19585550101",
    "deliveryStatus": "MessageWaiting"
  },
  {
    "address": "tel:+19585550104",
    "deliveryStatus": "MessageWaiting"
  }
  ]
}
D.22 Read delivery notification subscriptions (section 6.10.3.1)

Request:

GET /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions HTTP/1.1
Accept: application/json
Host: example.com

Response:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
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://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000"
  },
  {"callbackReference": {
    "callbackData": "54321",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
    "filterCriteria": "0103",
    "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000"
  }
],
"resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/"
}

D.23 Create outbound delivery notification subscription, using tel URI (section 6.10.5.1)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"deliveryReceiptSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
  "filterCriteria": "0102",
  "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000"
}}
Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/66666"
   },
   "filterCriteria": "0102",
   "resourceURL": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550100/subscriptions/sub000"
}}

D.24 Create outbound delivery notification subscription, using ACR (section 6.10.5.2)

Request:

POST /exampleAPI/smsmessaging/v1/outbound/acr%3Apseudonym123/subscriptions HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: example.com

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/66666"
   },
   "filterCriteria": "0102"
}}

Response:

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/smsmessaging/v1/outbound/acr%3Apseudonym123/subscriptions/sub000
Content-Length: nnnn
Date: Thu, 04 Jun 2009 02:51:59 GMT

{"deliveryReceiptSubscription": {
   "callbackReference": {
      "callbackData": "12345",
      "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification/66666"
   },
   "filterCriteria": "0102"
}}
D.25 Read delivery notification subscription (section 6.11.3.1)

Request:

GET /exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000 HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

{"deliveryReceiptSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/DeliveryInfoNotification"
  },
  "filterCriteria": "0102",
  "resourceURL": "http://example.com/exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000"}
}

D.26 Delete subscription for a client (section 6.11.6.1)

Request:

DELETE /exampleAPI/smssmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000 HTTP/1.1
Accept: application/json
Host: example.com

Response:

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

D.27 Notify client about message delivery status (section 6.12.5.1)

Request:

POST /notifications/DeliveryInfoNotification HTTP/1.1
Accept: application/json
Content-Type: application/json
Content-Length: nnnn
Host: application.example.com

{"deliveryInfoNotification": {
    "callbackData": "12345",
    "deliveryInfo": {
        "address": "tel:+19585550101",
        "deliveryStatus": "DeliveredToNetwork"
    },
    "link": {
        "href": "http://example.com/exampleAPI/smsmessaging/v1/outbound/tel%3A%2B19585550151/subscriptions/sub000",
        "rel": "DeliveryReceiptSubscription"
    }
}}

Response:

HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2009 02:51:59 GMT
## Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/methods and Parlay X [3GPP 29.199-4] equivalent operations.

<table>
<thead>
<tr>
<th>REST Resource</th>
<th>REST Method</th>
<th>REST 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>6.1.3</td>
<td>getReceivedSms¹</td>
</tr>
<tr>
<td>Inbound SMS messages retrieve and delete using registration</td>
<td>POST</td>
<td>6.2.5</td>
<td>getReceivedSms</td>
</tr>
<tr>
<td>Inbound SMS message subscriptions</td>
<td>POST</td>
<td>6.4.5</td>
<td>startSmsNotification</td>
</tr>
<tr>
<td>Individual inbound SMS message subscription</td>
<td>DELETE</td>
<td>6.5.6</td>
<td>stopSmsNotification</td>
</tr>
<tr>
<td>Client notification about inbound SMS message</td>
<td>POST</td>
<td>6.6.5</td>
<td>notifySMSReception</td>
</tr>
<tr>
<td>Outbound SMS message requests</td>
<td>POST</td>
<td>6.7.5</td>
<td>sendSms</td>
</tr>
<tr>
<td>Outbound SMS message delivery status</td>
<td>GET</td>
<td>6.9.3</td>
<td>getSmsDeliveryStatus</td>
</tr>
<tr>
<td>Outbound SMS message delivery notification subscriptions</td>
<td>POST</td>
<td>6.10.5</td>
<td>startDeliveryReceiptNotification</td>
</tr>
<tr>
<td>Individual outbound SMS message delivery notification subscription</td>
<td>DELETE</td>
<td>6.11.6</td>
<td>stopDeliveryReceiptNotification</td>
</tr>
<tr>
<td>Client notification about outbound SMS message delivery status</td>
<td>POST</td>
<td>6.12.5</td>
<td>notifySMSDeliveryReceipt</td>
</tr>
</tbody>
</table>

Table 1: Parlay X operations mapping

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).
Appendix F. Light-weight resources (Informative)

As this version of the specification does not define any light-weight resources, this appendix is empty.
Appendix G. Authorization aspects (Normative)

This appendix specifies how to use the RESTful Short Messaging API in combination with some authorization frameworks.

G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Short Messaging API MAY support the authorization framework defined in [Autho4API_10].

A RESTful Short Messaging API supporting [Autho4API_10]:

- SHALL conform to section D.1 of [REST_NetAPI_Common];
- SHALL conform to this section G.1.

G.1.1 Scope values

G.1.1.1 Definitions

In compliance with [Autho4API_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Short Messaging API:

- SHALL support the scope values defined in the table below;
- MAY support scope values not defined in this specification.

<table>
<thead>
<tr>
<th>Scope value</th>
<th>Description</th>
<th>For one-time access token</th>
</tr>
</thead>
<tbody>
<tr>
<td>oma_rest_sms.all_{apiVersion}</td>
<td>Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the “apiVersion” URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.</td>
<td>No</td>
</tr>
<tr>
<td>oma_rest_sms.in_regist</td>
<td>Provide access to all defined operations on inbound SMS messages using registration</td>
<td>No</td>
</tr>
<tr>
<td>oma_rest_sms.in_subscr</td>
<td>Provide access to all defined operations on inbound SMS messages using subscription</td>
<td>No</td>
</tr>
<tr>
<td>oma_rest_sms.out</td>
<td>Provide access to all defined operations on outbound SMS messages</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Scope values for RESTful Short Messaging API

G.1.1.2 Downscoping

In the case where the client requests authorization for “oma_rest_sms.all_{apiVersion}” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma_rest_sms.in_regist”
- “oma_rest_sms.in_subscr”
• “oma_rest_sms.out”

G.1.1.3 Mapping with resources and methods

Tables in this section specify how the scope values defined in section G.1.1.1 for the RESTful Short Messaging API map to the REST resources and methods of this API. In these tables, the root “oma_rest_sms.” of scope values is omitted for readability reasons.
### Table 3: Required scope values for: Inbound SMS messages for periodic polling

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Section reference</th>
<th>HTTP verbs</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound SMS messages for a given registration</td>
<td>/inbound/registrations/{registrationId}/messages</td>
<td>6.1</td>
<td>all_{apiVersion} or in_regist</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Inbound SMS messages retrieve and delete using registration</td>
<td>/inbound/registrations/{registrationId}/messages/retrieveAndDeleteMessages</td>
<td>6.2</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>all_{apiVersion} or in_regist</td>
<td>n/a</td>
</tr>
<tr>
<td>Inbound SMS message for a given registration</td>
<td>/inbound/registrations/{registrationId}/messages/{messageId}</td>
<td>6.3</td>
<td>all_{apiVersion} or in_regist</td>
<td>n/a</td>
<td>n/a</td>
<td>all_{apiVersion} or in_regist</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### Table 4: Required scope values for: Subscription management for inbound SMS messages

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Section reference</th>
<th>HTTP verbs</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound SMS message subscriptions</td>
<td>/inbound/subscriptions</td>
<td>6.4</td>
<td>all_{apiVersion} or in_subscr</td>
<td>n/a</td>
<td>all_{apiVersion} or in_subscr</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Individual inbound SMS message subscription</td>
<td>/inbound/subscriptions/{subscriptionId}</td>
<td>6.5</td>
<td>all_{apiVersion} or in_subscr</td>
<td>all_{apiVersion} or in_subscr</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>

© 2013 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
### Table 5: Required scope values for: Sending SMS message and obtaining the delivery status

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Section reference</th>
<th>HTTP verbs</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound SMS message requests</td>
<td>/outbound/{senderAddress}/requests</td>
<td>6.7</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Outbound SMS message request and delivery status</td>
<td>/outbound/{senderAddress}/requests/{requestId}</td>
<td>6.8</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td></td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Outbound SMS message delivery status</td>
<td>/outbound/{senderAddress}/requests/{requestId}/deliveryInfos</td>
<td>6.9</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td></td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### Table 6: Required scope values for: Subscription management for outbound SMS message delivery status

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL Base URL: http://{serverRoot}/smsmessaging/{apiVersion}</th>
<th>Section reference</th>
<th>HTTP verbs</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound SMS message delivery notification subscriptions</td>
<td>/outbound/{senderAddress}/subscriptions</td>
<td>6.10</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Individual outbound SMS message delivery notification subscription</td>
<td>/outbound/{senderAddress}/subscriptions/{subscriptionId}</td>
<td>6.11</td>
<td>all_{apiVersion} or out</td>
<td>n/a</td>
<td></td>
<td>n/a</td>
<td>all_{apiVersion} or out</td>
</tr>
</tbody>
</table>
G.1.2 Use of ‘acr:auth’

This section specifies the use of ‘acr:auth’ in place of an end user identifier in a resource URL path.

An ‘acr’ URI of the form ‘acr:auth’, where ‘auth’ is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

A client MAY use ‘acr:auth’ in a resource URL in place of the {senderAddress} resource URL variable in the resource URL path, when the RESTful Short Messaging API is used in combination with [Autho4API_10].

In the case the RESTful Short Messaging API supports [Autho4API_10], the server:

- **SHALL** accept ‘acr:auth’ as a valid value for the resource URL variable {senderAddress}.

**SHALL** conform to [REST_Common_TS] section 5.8.1.1 regarding the processing of ‘acr:auth’