Telia Monikanavaviestintä

SMS Messaging

Web Service API

Version 2.0

Powered by
LEKAB
Communication Systems

Table of Contents

Introduction

Background

A Web service is a standardized way of integrating Web-based applications using XML and other standards. Web services allow different applications from different sources to communicate with each other without timeconsuming custom coding. The "web services" model uses WSDL and SOAP with WS-Security.

The Telia Monikanavaviestintä service (MKV) Messaging WebService API facilitates sending and receiving SMS and MMS messages and querying sent messages for delivery status.

Consuming the web services

There are many ways to consume Web Services.

The first way is to use a SOAP Proxy Client Object generated by the WSDL utility, and it provides programmers with their familiar object model that they can use to call methods provided by the generated Proxy Interface.

The second way is to use HTTP-POST and HTTP-GET protocols.

This document covers the details of consuming web-services the first way, i.e. SOAP Proxy Client Object generated from a WSDL.

Web Services Description Language (WSDL) is an XML based protocol for information exchange in decentralized and distributed environments. A WSDL document defines services as collections of network endpoints, or ports. A WSDL document uses the following elements in the definition of network services:

- Types – a container for data type definitions using some type system (such as XSD).
- Message – an abstract, typed definition of the data being communicated.
- Operation – an abstract description of an action supported by the service.
- Port Type – an abstract set of operations supported by one or more endpoints.
- Binding – a concrete protocol and data format specification for a particular port type.
- Port – a single endpoint defined as a combination of a binding and a network address.
- Service – a collection of related endpoints.

Functional overview

The MKV system provides the following functionality for SMS and MMS:

- Sending mobile terminated (MT) short messages, such as text messages, OneWay and TwoWay SMS and MMS
- Receiving delivery reports for submitted MT messages
- Receiving mobile originated (MO) short messages TwoWay SMS, SMS to shortcode and MMS to shortcode.

For MO messages, MKV routes messages to a specific inbox based on:

- Message text pattern
- Destination address
- Any combination of the above

SMS two way messages

TwoWay SMS enables the user to send a SMS to a recipient from a computer and receive a reply into the same computer/user account.

Web Service Client

MKV provides API:s exposed as a web service with a SOAP interface. The SOAP protocol and the MKV server are independent of the platform used on the client side, although the installation of the SOAP tools could be different. The web service API is described in WSDL. For those not familiar with web services, MKV also provides a set of Java classes generated from the web service WSDL description upon request. These classes can be used as an SDK and are provided by MKV.

Web service

The web service URL and the location of the WSDL file is provided by MKV at the following address:

https://ws.mkv.telia.fi/ws/messaging-v2.wsdl

You can also find the WSDL in the appendix.

Security

Sending/receiving messages:

For authentication, the user ID and password are submitted in every web service invocation. It is the responsibility of the User to keep this user ID and password protected. Each message requires a UsernameToken WS-Security header with passwords of type PasswordText and a Timestamp WS-Security header.

For connection security the usage of HTTPS when accessing the web services is mandatory. The MKV server certificate is signed by Telia CA.

WS-Security header example

```
<soapenv:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-3" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>YourUsername</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">YourPassword</wsse:Password>
    </wsse:UsernameToken>
    <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-426">
      <wsu:Created>2012-02-03T16:14:09.938Z</wsu:Created>
      <wsu:Expires>2012-02-03T16:19:09.938Z</wsu:Expires>
    </wsu:Timestamp>
  </wsse:Security>
</soapenv:Header>
```

Error handling

Errors are received as SOAP faults. The SOAP fault contains the error details in the detail section of the SOAP fault.

SOAP fault example

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-426">
        <wsu:Created>2012-02-03T16:14:09.938Z</wsu:Created>
        <wsu:Expires>2012-02-03T16:19:09.938Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring xml:lang="en">VALIDATION ERROR</faultstring>
      <detail>
        <errorDetails xmlns="http://www.lekab.com/schema/messaging-v2/messages">
          <errorDetail>
            <errorCode>101</errorCode>
            <errorDescription>Sender required</errorDescription>
          </errorDetail>
          <errorDetail>
            <errorCode>102</errorCode>
            <errorDescription>At least one recipient is required</errorDescription>
          </errorDetail>
        </errorDetails>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Read timeout

Since invocations on MKV APIs normally results in MKV invoking other external systems, such as SMSCs, we recommend the user to use a rather high read timeout. We recommend 2 minutes read timeout for HTTP requests.

Connecting to MKV Web Services

This checklist presents an overview of the activities needed to connect to the MKV Web Services:

- Make an agreement with MKV.
- Contact MKV sales for activation of accounts and passwords.
- Integrate with the necessary MKV web service(s).
- Verify the integration towards MKV before deployment

Mobile terminated message integration with MKV

The Web Service API has three available operations:

- Send
- GetMessageStatus
- GetIncomingMessages.

SOAP examples can be found in the Examples section.

Send

The Send operation is used to send SMS and MMS messages.

SendRequest

The following parameters used for sending SMS and MMS messages.

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//SendRequest /sender	string	Mandatory/ Optional		11/15	The sender id. Max length 11 if alphanumeric and 15 if an MSISDN or shortcode. Mandatory for one way SMS and MMS (shortcode). Optional for replyable SMS (numbers provided by MKV)
//SendRequest /recipients /recipient	string	Mandatory		15	The MSISDN of the recipient. Must be the recipient phone number including country code e.g. 46706352602. At least one recipient is required. Up to 1000 recipients is supported by the interface, but actual maximum differs based on recipients and used operator and can be remarkable lower. For mass sendings there exists a different webservice interface. Contact sales for more info.

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//SendRequest /replyable	boolean	Mandatory	false		Set to true for Replyable messages. Only applicable for SMS.
//SendRequest /conversationId	string	Optional		256	User supplied conversation id. If the message is replyable the reply will get this conversationId.
//SendRequest /priority	enumeration	Optional	Normal		One of: Normal High Low
//SendRequest /scheduledDelivery	dateTime	Optional	Immediate		The time for scheduled delivery. Must be in the future.
//SendRequest /validTo	dateTime	Optional	7 days		The time a message is valid to before being removed from the send queue. Must be in the future.
//SendRequest /tariff	string	Optional		40	Mandatory for Premium messages.
//SendRequest /contentCategory	string	Optional			Mandatory for Premium messages.
//SendRequest /contentMetaData	string	Unsupported			Currently unsupported.
//SendRequest /vat	double	Optional			The VAT to apply for this message.
//SendRequest /referenceId	string	Optional		150	The id of the incoming message. Mandatory for Premium messages.
//SendRequest /attributes		Optional			This is used for special cases where extra attributes are used for the request. Not for normal use.
//SendRequest /data /sms or //SendRequest /data /mms	choice	Mandatory			The data of the message. Must be one of sms or mms
SMS Specific Options					

//Element /@attribute	**Type**	**Mandatory/ Optional/ Unsupported**	**Default value**	**Max Length**	**Description**
//SendRequest /data /sms /payload /message	byte[] (base64Binary)	Mandatory		804	The message data if a text message this will be an UTF-8 encoded string converted to a byte array
//SendRequest /data /sms /payload /udh	byte[] (base64Binary)	Unsupported			Currently unsupported. The service currently auto splits messages longer than 160 characters
//SendRequest /data /sms /encoding	integer	Unsupported			Currently unsupported. The service autodetects the encoding for text messages.
//SendRequest /data /sms /replaceIfPresent	integer	Optional			Try to replace the message at the mobile station.
//SendRequest /data /sms /flash	boolean	Optional	false		Send the message as a flash SMS.
//SendRequest /data /sms /port	integer	Unsupported			Currently Unsupported. Address the message at the mobile station. E.g. to an application.
//SendRequest /data /sms /attributes		Unsupported			Reserved for future use
MMS Specific Options					
//SendRequest /data /mms /payload /subject	string	Optional		500	The subject of the MMS
//SendRequest /data /mms /payload /attachments	list	Mandatory			At least one attachment has to be included.

//Element /@attribute	**Type**	**Mandatory/ Optional/ Unsupported**	**Default value**	**Max Length**	**Description**
//SendRequest /data /mms /payload /attachments /attachment /name	string	Mandatory		255	The name of the attachment
//SendRequest /data /mms /payload /attachments /attachment /contentType	string	Mandatory		255	The content type of the attachment
//SendRequest /data /mms /payload /attachments /attachment /data	byte[] (base64Binary)	Mandatory			The binary data of the attachment
//SendRequest /data /mms /attributes		Unsupported			Reserved for future use.

SendResponse

A list of messageStatus objects is returned. One for each recipient in the send list.

//Element /@attribute	Type	Max Length	Description
//SendResponse /messageStatus /statusCode	integer		The status code of the message
//SendResponse /messageStatus /statusText	string	100	The status text for the given status code.
//SendResponse /messageStatus /id	string	150	The message ID (use this message id for the MessageStatus request)
//SendResponse /messageStatus /sender	string	15	The sender of the message
//SendResponse /messageStatus /recipient	string	15	The recipient of the message
//SendResponse /messageStatus /time	dateTime		The timestamp of the status
//SendResponse /messageStatus /billingStatus	integer		The billing status of the message. Only applicable for premium messages
//SendResponse /messageStatus /attributes			Extra attributes describing the message such as NumberOfMessages and NumberOfCharacters
//SendResponse /attributes			Extra attributes describing the send request such as TotalNumberOfMessages and QuotaLeft

Error codes

The following error codes can be received in the SOAP fault when sending messages.

Error Code	Error Text	Description
1	System Error	A System Error occurred.
2	Send Failed	Send failed
10	Access Denied	Access Denied. If the username and password are incorrect or the user does not have the appropriate authorization role.
11	Quota Exceeded	The account quota is exceded (Only returned if the account has an attached monthly quota)
12	Unknown User	The user is unknown (Not returned for normal use)
13	Unknown Server	The server is unknown (Not returned for normal use)
100	Validation Error for field "{}": "{}"	General validation error. {} is replaced with the field name and error description respectively.
101	"{}" is not a valid sender	Invalid sender. {} is replaced with the sender
101	Sender Required	Sender is required for this message

Error Code	Error Text	Description
102	"{}" is not a valid recipient	Invalid recipient. {} is replaced with the invalid recipient.
102	At least one recipient is required	At least one recipient is required
103	Tariff is Mandatory for Premium SMS	Tariff is Mandatory for Premium SMS
104	Reference Id is Mandatory for Premium SMS	Reference Id is Mandatory for Premium SMS
105	Invalid Message Length	The length of the SMS message is invalid
106	Invalid Subject Length	The length of the MMS subject is invalid
107	Content Category Mandatory for Premium Messages	Content Category Mandatory for Premium Messages
108	Content Category Invalid	Content Category Invalid
109	Scheduled Delivery Invalid	The scheduled delivery time is not a valid date or is not in the future.
114	Reference Id Not Found	The reference id specified is not found. (Only applicable for premium messages)
115	Valid To Invalid	The validity period is not a valid date or is not in the future.
116	Conversation Id Invalid	The conversation id is not a valid conversation id.

Message status codes

The following message status codes can be received in the message status.

Status Code	Status Text	Description
0	QUEUED	Queued for delivery
1	SENT	Sent to operator
2	DELIVERED	Delivered to the mobile station
3	DELETED	The message was deleted
4	EXPIRED	The message has expired
5	REJECTED	The message was rejected by the operator
6	UNDELIVERABLE	The message could not be delivered
7	ACCEPTED	The message was accepted by the operator
8	ABSENT SUBSCRIBER	The subscribers mobile station is switched off
9	UNKNOWN SUBSCRIBER	The subscriber is not known
10	INVALID DESTINATION	The destination address is invalid
11	SUBSCRIBER ERROR	The mobile station can not receive the message
12	UNKNOWN	The status of the message is unknown
13	ERROR	Internal error when sending the message

GetMessageStatus

The get message status operation is used for querying the MKV system for the message send status.

GetMessageStatusRequest

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//GetMessageStatusRequest /@markStatusesRead	boolean	Mandatory	false		Mark the retrieved messages as read
//GetMessageStatusRequest /@maxNumberOfStatuses	integer	Mandatory	100		Number of message statuses to retrieve (not applicable if message ids specified)
//GetMessageStatusRequest /messageIds /messageId	string	Optional		150	Supply a list of message ids for retrieving message statuses or marking message statuses as read.
//GetMessageStatusRequest /attributes		Optional			This is used for special cases where extra attributes are used for the request. Not for normal use.

GetMessageStatusResponse

//Element /@attribute	Type	Max Length	Description
//GetMessageStatusResponse /messageStatus /statusCode	integer		The status code of the message
//GetMessageStatusResponse /messageStatus /statusText	string	100	The status text for the given status code.
//GetMessageStatusResponse /messageStatus /id	string	150	The message ID (use this message id for the MessageStatus request)
//GetMessageStatusResponse /messageStatus /sender	string	15	The sender of the message
//GetMessageStatusResponse /messageStatus /recipient	string	15	The recipient of the message
//GetMessageStatusResponse /messageStatus /time	dateTime		The timestamp of the status
//GetMessageStatusResponse /messageStatus /billingStatus	integer		The billing status of the message. Only applicable for premium messages
//GetMessageStatusResponse /messageStatus /attributes			Extra attributes describing the message such as NumberOfMessages and NumberOfCharacters
//GetMessageStatusResponse /attributes			Extra attributes returned e.g. QuotaLeft

Error codes

The following error codes can be received in the SOAP fault when querying message status.

Error Code	Error Text	Description
1	System Error	A System Error occurred.
10	Access Denied	Access Denied. If the username and password are incorrect or the user does not have the appropriate authorization role.
12	Unknown User	The user is unknown (Not returned for normal use)
13	Unknown Server	The server is unknown (Not returned for normal use)
100	Validation Error for field "{}": "{}"	General validation error. {} is replaced with the field name and error description respectively.
110	Invalid Id	The ID specified is invalid
111	No Message Found for Id	The message with the specified id can not be found
112	No Ids Specified	No ids specified
113	No Message Found	No message found
130	Invalid number of statuses to retreive	Invalid number of statuses to retrieve

Error Code	Error Text	Description
131	Invalid number of status ids. Min 1, Max 1000	Invalid number of status ids specified

Message status codes

The following message status codes can be received for the queried messages.

Status Code	Status Text	Description
0	QUEUED	Queued for delivery
1	SENT	Sent to operator
2	DELIVERED	Delivered to the mobile station
3	DELETED	The message was deleted
4	EXPIRED	The message has expired
5	REJECTED	The message was rejected by the operator
6	UNDELIVERABLE	The message could not be delivered
7	ACCEPTED	The message was accepted by the operator
8	ABSENT SUBSCRIBER	The subscribers mobile station is switched off
9	UNKNOWN SUBSCRIBER	The subscriber is not known
10	INVALID DESTINATION	The destination address is invalid
11	SUBSCRIBER ERROR	The mobile station can not receive the message
12	UNKNOWN	The status of the message is unknown
13	ERROR	Internal error when sending the message

GetIncomingMessages

This operation is used for retrieving incoming messages and marking messages as retrieved.

GetIncomingMessagesRequest

//Element /@attribute	**Type**	**Mandatory/ Optional/ Unsupported**	**Default value**	**Max Length**	**Description**
//GetIncomingMessagesRequest /@markMessagesRead	boolean	Mandatory	false		Mark the retrieved messages as read
//GetIncomingMessagesRequest /@maxNumberOfMessages	integer	Mandatory	10		Number of messages to retrieve (not applicable if message ids specified)
//GetIncomingMessagesRequest /@retreiveMessages	boolean	Mandatory	true		Retrieve messages. Set this to false if the data of the messages should not be retrieved. E.g. when marking already retrieved messages as read.
//GetIncomingMessagesRequest /messageIds /messageId	string	Optional		150	Supply a list of message ids for retrieving messages or marking messages as read.
//GetIncomingMessagesRequest /attributes		Optional			This is used for special cases where extra attributes are used for the request. Not for normal use.

GetIncomingMessagesResponse

//Element /@attribute	Type	Max Length	Description
//GetIncomingMessagesResponse /attributes			Extra attributes returned e.g. QuotaLeft
//GetIncomingMessagesResponse /incomingMessages /id	string	150	The id of the message
//GetIncomingMessagesResponse /incomingMessages /sender	string	15	The sender of the message
//GetIncomingMessagesResponse /incomingMessages /recipient	string	15	The recipient of the message
//GetIncomingMessagesResponse /incomingMessages /timeStamp	dateTime		The time the message was received
//GetIncomingMessagesResponse /incomingMessages /conversationId	string	50	The conversationId of the message (applicable if this was a reply to a replyable outgoing message)
//GetIncomingMessagesResponse /incomingMessages /outgoingMessageId	string	150	The ID of the outgoing message (if applicable)
//GetIncomingMessagesResponse /incomingMessages /payload /sms or //GetIncomingMessagesResponse /incomingMessages /payload /mms			The Payload of the message. One of SMS payload or MMS payload
//GetIncomingMessagesResponse /incomingMessages /outgoingMessagePayload /sms or //GetIncomingMessagesResponse /incomingMessages /outgoingMessagePayload /mms			The payload of the outgoing message. (if applicable) One of SMS payload or MMS payload
SMS Specific Options			
//GetIncomingMessagesResponse /incomingMessages /payload /sms /message	byte[] (base64Binary)		The message a UTF-8 encoded string converted to a byte[]
//GetIncomingMessagesResponse /incomingMessages /payload	byte[] (base64Binary)		Unsupported

/sms/udh			
//GetIncomingMessagesResponse/incomingMessages/outgoingMessagePayload/sms/message	byte[] (base64Binary)		The message a UTF-8 encoded string converted to a byte[]
//GetIncomingMessagesResponse/incomingMessages/outgoingMessagePayload/sms/udh	byte[] (base64Binary)		Unsupported
MMS Specific Options			
//GetIncomingMessagesResponse/incomingMessages/payload/mms/subject	string	500	The subject of the MMS
//GetIncomingMessagesResponse/incomingMessages/payload/mms/attachments	list		At least one attachment is included
//GetIncomingMessagesResponse/incomingMessages/payload/mms/attachments/attachment/name	string	255	The name of the attachment
//GetIncomingMessagesResponse/incomingMessages/payload/mms/attachments/attachment/contentType	string	255	The content type of the attachment
//GetIncomingMessagesResponse/incomingMessages/payload/mms/attachments/attachment/data	byte[] (base64Binary)		The binary data of the attachment
//GetIncomingMessagesResponse/incomingMessages/outgoingMessagePayload/mms/subject	string	500	The subject of the MMS
//GetIncomingMessagesResponse/incomingMessages/outgoingMessagePayload/mms/attachments	list		At least one attachment is included
//GetIncomingMessagesResponse/incomingMessages	string	255	The name of the attachment

/outgoingMessagePayload /mms /attachments /attachment /name			
//GetIncomingMessagesResponse /incomingMessages /outgoingMessagePayload /mms /attachments /attachment /contentType	string	255	The content type of the attachment
//GetIncomingMessagesResponse /incomingMessages /outgoingMessagePayload /mms /attachments /attachment /data	byte[] (base64Binary)		The binary data of the attachment

Error codes

The following error codes can be received when retrieving incoming messages.

Error Code	Error Text	Description
1	System Error	A System Error occurred.
10	Access Denied	Access Denied. If the username and password are incorrect or the user does not have the appropriate authorization role.
12	Unknown User	The user is unknown (Not returned for normal use)
13	Unknown Server	The server is unknown (Not returned for normal use)
100	Validation Error for field "{}": "{}"	General validation error. {} is replaced with the field name and error description respectively.
110	Invalid Id	The ID specified is invalid
111	No Message Found for Id	The message with the specified id can not be found
112	No Ids Specified	No ids specified
113	No Message Found	No message found
140	Invalid number of messages to retrieve	Invalid number of messages to retrieve
141	Invalid number of message ids. Min 1, Max 100	Invalid number of message ids specified

Examples

The examples are in SOAP format only. For code examples in for example Java, C# .Net, PHP or C etc. Contact MKV sales. The SOAP Examples are formatted for readability. The examples are made with SoapUI http://www.soapui.org/ .

Sending a one way SMS

Request

```
<soapenv:Envelope xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>YourUsername</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">YourPassword</wsse:Password>
      </wsse:UsernameToken>
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <mes:SendRequest>
      <mes:sender>Lekab</mes:sender>
      <mes:recipients>
        <mes:recipient>46706352602</mes:recipient>
      </mes:recipients>
      <mes:replyable>false</mes:replyable>
      <mes:data>
        <mes:sms>
          <mes:payload>
            <mes:message>SGVsbG8gV29ybGQK</mes:message>
          </mes:payload>
        </mes:sms>
      </mes:data>
    </mes:SendRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <mes:SendResponse xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages">
      <mes:messageStatus>
        <mes:statusCode>0</mes:statusCode>
        <mes:statusText>QUEUED</mes:statusText>
        <mes:id>1-1282358</mes:id>
        <mes:sender>Lekab</mes:sender>
        <mes:recipient>46706352602</mes:recipient>
        <mes:time>2010-05-07T09:00:07.723+02:00</mes:time>
        <mes:billingStatus>0</mes:billingStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>NumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
          <mes:attribute>
            <mes:name>NumberOfCharacters</mes:name>
            <mes:value><mes:integer>11</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
      </mes:messageStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>TotalNumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
    </mes:SendResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sending a replyable two way SMS

Request

```
<soapenv:Envelope xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>YourUsername</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">YourPassword</wsse:Password>
      </wsse:UsernameToken>
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <mes:SendRequest>
      <mes:recipients>
        <mes:recipient>46706352602</mes:recipient>
      </mes:recipients>
      <mes:replyable>true</mes:replyable>
      <mes:data>
        <mes:sms>
          <mes:payload>
            <mes:message>SGVsbG8gV29ybGQK</mes:message>
          </mes:payload>
        </mes:sms>
      </mes:data>
    </mes:SendRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <mes:SendResponse xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages">
      <mes:messageStatus>
        <mes:statusCode>0</mes:statusCode>
        <mes:statusText>QUEUED</mes:statusText>
        <mes:id>1-1282359</mes:id>
        <mes:sender>46737494333567</mes:sender>
        <mes:recipient>46706352602</mes:recipient>
        <mes:time>2010-05-07T09:06:30.027+02:00</mes:time>
        <mes:billingStatus>0</mes:billingStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>NumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
          <mes:attribute>
            <mes:name>NumberOfCharacters</mes:name>
            <mes:value><mes:integer>11</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
      </mes:messageStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>TotalNumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
    </mes:SendResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Getting message status of specified messages

Request

```
<soapenv:Envelope xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>YourUsername</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">YourPassword</wsse:Password>
      </wsse:UsernameToken>
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <mes:GetMessageStatusRequest markStatusesRead="false" maxNumberOfStatuses="100">
      <mes:messageIds>
        <mes:messageId>1-1282358</mes:messageId>
        <mes:messageId>1-1282359</mes:messageId>
      </mes:messageIds>
    </mes:GetMessageStatusRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <mes:GetMessageStatusResponse xmlns:mes="http://www.lekab.com/schema/messaging-
v2/messages">
      <mes:messageStatus>
        <mes:statusCode>2</mes:statusCode>
        <mes:statusText>DELIVERED</mes:statusText>
        <mes:id>1-1282358</mes:id>
        <mes:sender>Lekab</mes:sender>
        <mes:recipient>46706352602</mes:recipient>
        <mes:time>2010-05-07T09:00:08.000+02:00</mes:time>
        <mes:billingStatus>0</mes:billingStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>NumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
          <mes:attribute>
            <mes:name>NumberOfCharacters</mes:name>
            <mes:value><mes:integer>53</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
      </mes:messageStatus>
      <mes:messageStatus>
        <mes:statusCode>2</mes:statusCode>
        <mes:statusText>DELIVERED</mes:statusText>
        <mes:id>1-1282359</mes:id>
        <mes:sender>46737494333567</mes:sender>
        <mes:recipient>46706352602</mes:recipient>
        <mes:time>2010-05-07T09:06:50.000+02:00</mes:time>
        <mes:billingStatus>0</mes:billingStatus>
        <mes:attributes>
          <mes:attribute>
            <mes:name>NumberOfMessages</mes:name>
            <mes:value><mes:integer>1</mes:integer></mes:value>
          </mes:attribute>
          <mes:attribute>
            <mes:name>NumberOfCharacters</mes:name>
            <mes:value><mes:integer>53</mes:integer></mes:value>
          </mes:attribute>
        </mes:attributes>
      </mes:messageStatus>
    </mes:GetMessageStatusResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Getting incoming messages

Request

```
<soapenv:Envelope xmlns:mes="http://www.lekab.com/schema/messaging-v2/messages"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>YourUsername</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">YourPassword</wsse:Password>
      </wsse:UsernameToken>
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <mes:GetIncomingMessagesRequest markMessagesRead="false" maxNumberOfMessages="100"
retrieveMessages="true">
    </mes:GetIncomingMessagesRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-1852">
        <wsu:Created>2012-02-06T14:59:33.873Z</wsu:Created>
        <wsu:Expires>2012-02-06T15:04:33.873Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <mes:GetIncomingMessagesResponse xmlns:mes="http://www.lekab.com/schema/messaging-
v2/messages">
      <mes:incomingMessages>
        <mes:id>1-73098732</mes:id>
        <mes:sender>46706352602</mes:sender>
        <mes:recipient>46737494333567</mes:recipient>
        <mes:timeStamp>2010-05-07T09:06:52.000+02:00</mes:timeStamp>
        <mes:payload>
          <mes:sms>
            <mes:message>
TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gQ3VyYWJpdHVyIH
ByZXRpdW0gbG9ib3J0aXMgbmVxdWUsIHZlaGljdWxhIGNvbmRpbWVudHVtIGFyY3UgbWF0dGlzIHVsdHJpY2llcy4g
U2VkIHF1YW0gdm9sdXRwYXQuTG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVyIGFkaXBpc2Npbm
cgZWxpdC4gVmVzdGlidWx1bSBuZWMgbG9yZW0gbm9uIGRpYW0gdGVtcG9yIHBvcnR0aXRvciBibGFuZGl0IHZlbCBu
aXNpLiBTdXNwZW5kaXNzZSBwb3RlbnRpIG1hc3NhIG51bmMu
            </mes:message>
          </mes:sms>
        </mes:payload>
        <mes:outgoingMessageId>1-1282359</mes:outgoingMessageId>
        <mes:outgoingMessagePayload>
          <mes:sms>
            <mes:message>SGVsbG8gV29ybGQK</mes:message>
          </mes:sms>
        </mes:outgoingMessagePayload>
      </mes:incomingMessages>
    </mes:GetIncomingMessagesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Appendix

WSDL

```
<?xml version="1.0" encoding="UTF-8"?><wsdl:definitions
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:schema="http://www.lekab.com/schema/messaging-v2/messages"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://www.lekab.com/schema/messaging-v2/messages">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.lekab.com/schema/messaging-v2/messages" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="http://www.lekab.com/schema/messaging-
v2/messages">
    <xsd:element name="SendRequest" type="OutgoingMessage"/>
    <xsd:element name="SendResponse" type="MessageStatuses"/>

    <xsd:element name="GetMessageStatusRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element maxOccurs="1" minOccurs="0" name="messageIds" type="MessageIds"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
            <xsd:attribute default="false" name="markStatusesRead" type="xsd:boolean"/>
            <xsd:attribute default="100" name="maxNumberOfStatuses" type="xsd:int"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="GetMessageStatusResponse" type="MessageStatuses"/>

    <xsd:element name="GetIncomingMessagesRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element maxOccurs="1" minOccurs="0" name="messageIds" type="MessageIds"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
            <xsd:attribute default="false" name="markMessagesRead" type="xsd:boolean"/>
            <xsd:attribute default="10" name="maxNumberOfMessages" type="xsd:int"/>
            <xsd:attribute default="true" name="retrieveMessages" type="xsd:boolean"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="GetIncomingMessagesResponse" type="IncomingMessages"/>
    <xsd:element name="errorDetails" type="ErrorDetails"/>

    <xsd:complexType name="OutgoingMessage">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="sender" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="recipients" type="Recipients"/>
            <xsd:element default="false" maxOccurs="1" minOccurs="1" name="replyable"
type="xsd:boolean"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="conversationId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="priority" type="Priority"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="scheduledDelivery"
type="xsd:dateTime"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="validTo" type="xsd:dateTime"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="tariff" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="contentCategory"
type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="contentMetaData"
type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="vat" type="xsd:double"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="referenceId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="data" type="Data"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="IncomingMessages">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="incomingMessages"
type="IncomingMessage"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="IncomingMessage">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="id" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="sender" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="recipient" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="conversationId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="timeStamp" type="xsd:dateTime"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="payload" type="Payload"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="outgoingMessageId"
type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="outgoingMessagePayload"
type="Payload"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="MessageIds">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="messageId"
type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="MessageStatuses">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="messageStatus"
type="MessageStatus"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="MessageStatus">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="statusCode" type="xsd:int"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="statusText" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="id" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="sender" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="recipient" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="conversationId" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="time" type="xsd:dateTime"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="billingStatus" type="xsd:int"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Data">
        <xsd:choice>
            <xsd:element name="sms" type="SmsData"/>
            <xsd:element name="mms" type="MmsData"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="Payload">
        <xsd:choice>
            <xsd:element name="sms" type="SmsPayload"/>
            <xsd:element name="mms" type="MmsPayload"/>
        </xsd:choice>
    </xsd:complexType>
    <xsd:complexType name="SmsPayload">
        <xsd:sequence>
            <xsd:element name="message" type="xsd:base64Binary"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="udh" type="xsd:base64Binary"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="MmsPayload">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="subject" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="attachments" type="Attachments"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="SmsData">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="payload" type="SmsPayload"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="encoding" type="xsd:int"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="replaceIfPresent" type="xsd:int"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="flash" type="xsd:boolean"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="port" type="xsd:int"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="MmsData">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="payload" type="MmsPayload"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="attributes" type="Attributes"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Attributes">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="attribute"
type="Attribute"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Attribute">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="name" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="value" type="Value"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Value">
        <xsd:choice>
            <xsd:element name="string" type="xsd:string"/>
            <xsd:element name="boolean" type="xsd:boolean"/>
            <xsd:element name="integer" type="xsd:int"/>
            <xsd:element name="long" type="xsd:long"/>
            <xsd:element name="float" type="xsd:float"/>
            <xsd:element name="double" type="xsd:double"/>
            <xsd:element name="base64Binary" type="xsd:base64Binary"/>
        </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="Attachments">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="attachment"
type="Attachment"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Attachment">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="name" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="contentType" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="data" type="xsd:base64Binary"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="Recipients">
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="recipient"
type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:simpleType name="Priority">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Normal"/>
            <xsd:enumeration value="High"/>
            <xsd:enumeration value="Low"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="ErrorDetails">
        <xsd:sequence>
            <xsd:sequence>
                <xsd:element maxOccurs="unbounded" minOccurs="1" name="errorDetail"
type="ErrorDetail"/>
            </xsd:sequence>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ErrorDetail">
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="errorCode" type="xsd:string"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="errorDescription"
type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
  </wsdl:types>
  <wsdl:message name="errorDetails">
    <wsdl:part element="schema:errorDetails" name="errorDetails">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetIncomingMessagesResponse">
    <wsdl:part element="schema:GetIncomingMessagesResponse"
name="GetIncomingMessagesResponse">
```

```
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetMessageStatusRequest">
    <wsdl:part element="schema:GetMessageStatusRequest" name="GetMessageStatusRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="SendResponse">
    <wsdl:part element="schema:SendResponse" name="SendResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetIncomingMessagesRequest">
    <wsdl:part element="schema:GetIncomingMessagesRequest" name="GetIncomingMessagesRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetMessageStatusResponse">
    <wsdl:part element="schema:GetMessageStatusResponse" name="GetMessageStatusResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="SendRequest">
    <wsdl:part element="schema:SendRequest" name="SendRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="messaging-v2">
    <wsdl:operation name="GetMessageStatus">
      <wsdl:input message="schema:GetMessageStatusRequest" name="GetMessageStatusRequest">
    </wsdl:input>
      <wsdl:output message="schema:GetMessageStatusResponse" name="GetMessageStatusResponse">
    </wsdl:output>
      <wsdl:fault message="schema:errorDetails" name="errorDetails">
    </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="GetIncomingMessages">
      <wsdl:input message="schema:GetIncomingMessagesRequest"
name="GetIncomingMessagesRequest">
    </wsdl:input>
      <wsdl:output message="schema:GetIncomingMessagesResponse"
name="GetIncomingMessagesResponse">
    </wsdl:output>
      <wsdl:fault message="schema:errorDetails" name="errorDetails">
    </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Send">
      <wsdl:input message="schema:SendRequest" name="SendRequest">
    </wsdl:input>
      <wsdl:output message="schema:SendResponse" name="SendResponse">
    </wsdl:output>
      <wsdl:fault message="schema:errorDetails" name="errorDetails">
    </wsdl:fault>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="messaging-v2Binding" type="schema:messaging-v2">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetMessageStatus">
      <soap:operation soapAction=""/>
      <wsdl:input name="GetMessageStatusRequest">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetMessageStatusResponse">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="errorDetails">
        <soap:fault name="errorDetails" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="GetIncomingMessages">
      <soap:operation soapAction=""/>
      <wsdl:input name="GetIncomingMessagesRequest">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="GetIncomingMessagesResponse">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="errorDetails">
        <soap:fault name="errorDetails" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="Send">
      <soap:operation soapAction=""/>
      <wsdl:input name="SendRequest">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="SendResponse">
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="errorDetails">
        <soap:fault name="errorDetails" use="literal"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="messaging-v2Service">
    <wsdl:port binding="schema:messaging-v2Binding" name="messaging-v2Port">
      <soap:address location="https://ws.mkv.sonera.fi:443/ws/messaging-v2"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```