Lekab Batch Web Service API v1

Lekab Communication Systems AB

Version 5.1.153, 2024-04-18

Lekab Batch Web Service API v1

Introduction . . . 1
1. Web Services . . . 2
1.1. SOAP . . . 2
1.2. WSDL . . . 4
1.2.1. Lekab WSDL URL . . . 5
1.3. Web Service Client . . . 5
1.4. Web Service Security . . . 5
1.5. Web Service Error Handling . . . 6
1.6. Read Timeout Setting . . . 7
1.7. How to Connect to Lekab Web Batch Services . . . 8
2. Batch Service API Operations . . . 9
2.1. "SendBatch" Operation . . . 9
2.1.1. SendBatchRequest . . . 9
2.1.2. SendBatchResponse . . . 13
2.2. "BatchInfo" Operation . . . 13
2.2.1. BatchInfoRequest . . . 14
2.2.2. BatchInfoResponse . . . 15
2.3. "BatchMessageId" Operation . . . 15
2.3.1. BatchMessageIdRequest . . . 15
2.3.2. BatchMessageIdResponse . . . 16
2.4. "BatchMessageStatus" Operation . . . 16
2.4.1. BatchMessageStatusRequest . . . 16
2.4.2. BatchMessageStatusResponse . . . 18
2.5. Message Status and Fault Handling . . . 19
2.5.1. Status Code . . . 20
2.5.2. Reason Code . . . 20
3. Appendices . . . 22
3.1. Batch Service Operation Examples . . . 22
3.1.1. Create a New SoapUI Project . . . 22
3.1.2. SendBatch . . . 27
3.1.3. BatchInfo . . . 29
3.1.4. BatchMessageId . . . 30
3.1.5. BatchMessageStatus . . . 31
3.2. URL Encoding Tool . . . 33
3.3. Base64 Encoding Tool . . . 34
3.4. Lekab Web Batch Service WSDL . . . 35

Introduction

© 2006 - 2024 Lekab Communication Systems AB. Version 5.1.153, 2024-04-18.

There is a great need of sending out a bulk of promotional messages, discount offers, seminars, new product information, event announcements, news alerts and many more applications. The SMS service is the perfect way to pro-actively reach out to the customers, the suppliers, the employees etc. With the Lekab Batch WS API you can harness the power of high volume SMS messaging by sending one text message to many mobile recipients. To make it simpler, all customized messages to several groups/single recipients can be combined into one file and sent out as one batch by Lekab Batch WS API. The Lekab Batch WS API can manage batches of limitless size.

This document describes the web service batch functions and usages in Lekab Virtual SMS Operator (VSO) system. The Lekab Messaging Web Service Batch API facilitates sending bulk SMS messages and querying sent messages for delivery status.

The Lekab system provides the following functionality for batch services in Figure 1, “Services”:

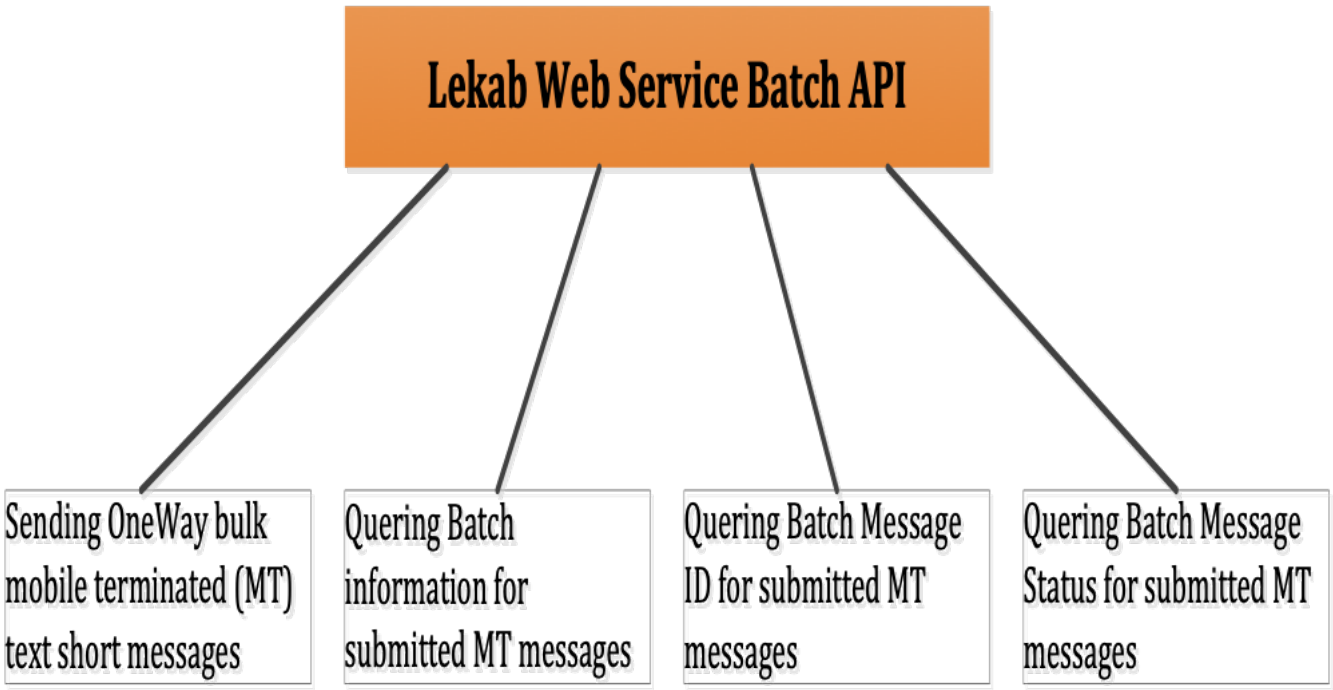

Figure 1. Services

The batch service examples made with SoapUI tool can be found in the Appendices.

Chapter 1. Web Services

A Web Service (WS) 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 time consuming custom coding. The "Web Services" model uses WSDL and SOAP with WS-Security.

1.1. SOAP

SOAP (Simple Object Access Protocol) [1] is the standard messaging protocol used by Web Services. SOAP's primary application is inter application communication. The SOAP structure is shown in Figure 2, "SOAP Envelope".

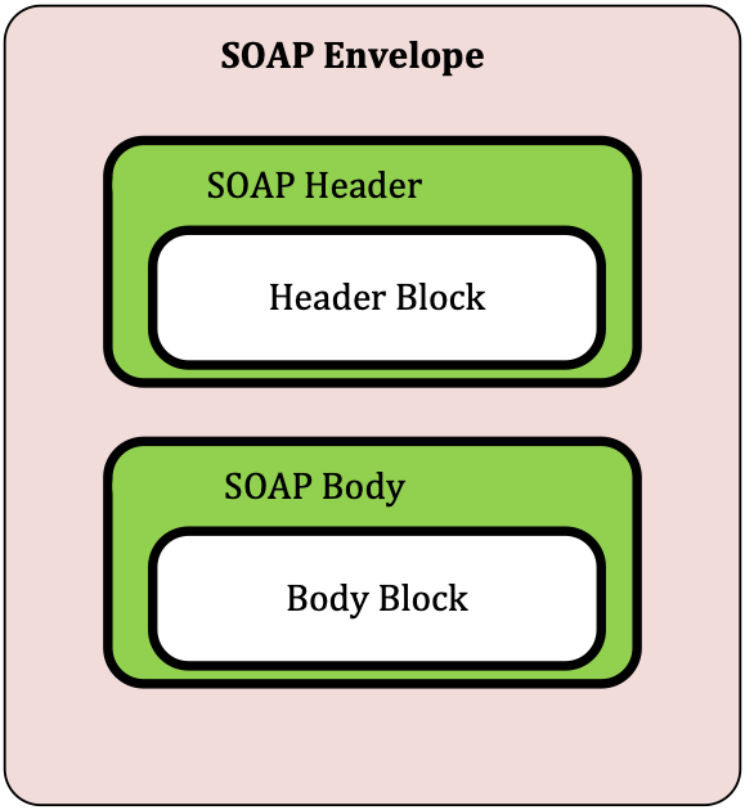

Figure 2. SOAP Envelope

SOAP codifies the use of XML as an encoding scheme for request and response parameters.

SOAP covers the following four main areas:

- A **message format** for one-way communication describing how a message can be packed into an XML document.
- A **description** of how a SOAP message should be transported using HTTP as a means for transport (for Web-based interaction) or SMTP (for e-mail-based interaction).
- A **set of rules** that must be followed when processing a SOAP message and a simple classification of the entities involved in processing a SOAP message.
- A **set of conventions** on how to turn an RPC call into a SOAP message and back.

An example of SOAP message is shown in Figure 3, “SOAP Example”.

```
<?xml version='1.0' ?>

<soapenv:Envelope xmlns:mes=http://www.lekab.com/schema/messaging/batch/v1/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"
         xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-28"><wsu:Created>2013-08-14T14:03:26Z</wsu:Created>
         <wsu:Expires>2013-08-14T14:13:26Z</wsu:Expires></wsu:Timestamp>
         <wsse:UsernameToken wsu:Id="UsernameToken-27"><wsse:Username>user1</wsse:Username>
         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">user
          </wsse:Password></wsse:UsernameToken>
      </wsse:Security>

    </soapenv:Header>

    <soapenv:Body>

      <mes:SendBatchRequest>
          <mes:sender>Lekab</mes:sender>
           <mes:recipients>
                <mes:recipient>4670123456788</mes:recipient>
                <mes:recipient>4670123456789</mes:recipient>
           </mes:recipients>
           <mes:message>SGVsbG8gd29ybGQ=LA==</mes:message>
      </mes:SendBatchRequest>

    </soapenv:Body>

</soapenv:Envelope>
```

Figure 3. SOAP Example

Figure 4, “Communication” shows how the client side communicates with the server side over HTTP protocol.

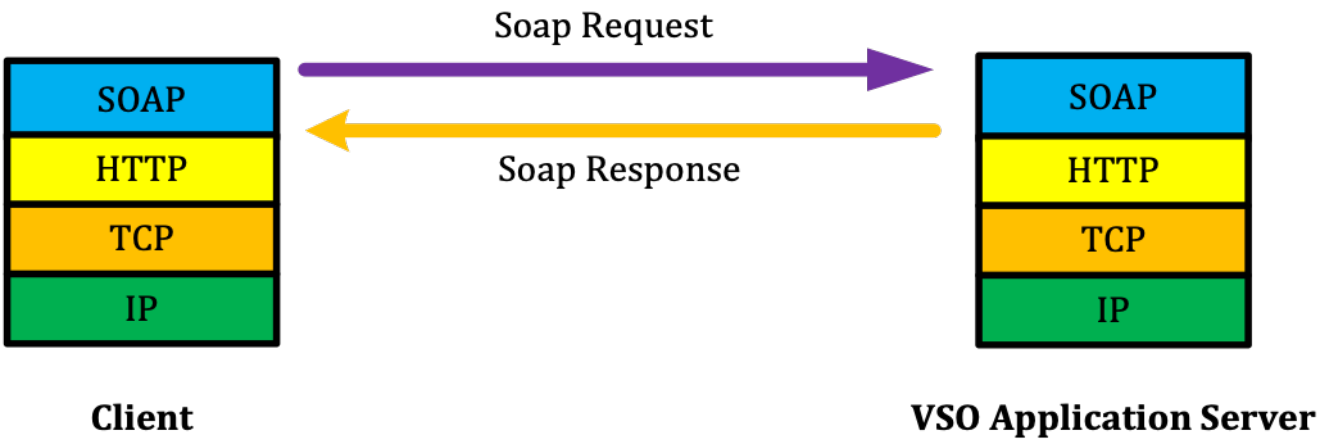

Figure 4. Communication

1.2. WSDL

There are many ways to consume Web Services:

- 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.
- Use HTTP-POST and HTTP-GET protocols.

This document covers the details of consuming web-services - SOAP Proxy Client Object generated from a WSDL.

The WSDL (Web Services Description Language) is an XML-based interface description language that is used for describing the functionality offered by a web service. 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:

ELEMENT	DESCRIPTION
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

1.2.1. Lekab WSDL URL

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

https://secure.lekab.com/batch/messaging-batch-v1.wsdl

You can also find the WSDL content for messaging batch in the Appendices.

1.3. Web Service Client

Lekab provides APIs exposed as a web service with a SOAP interface in Figure 5, "Web Service Client". The SOAP protocol and the Lekab 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.

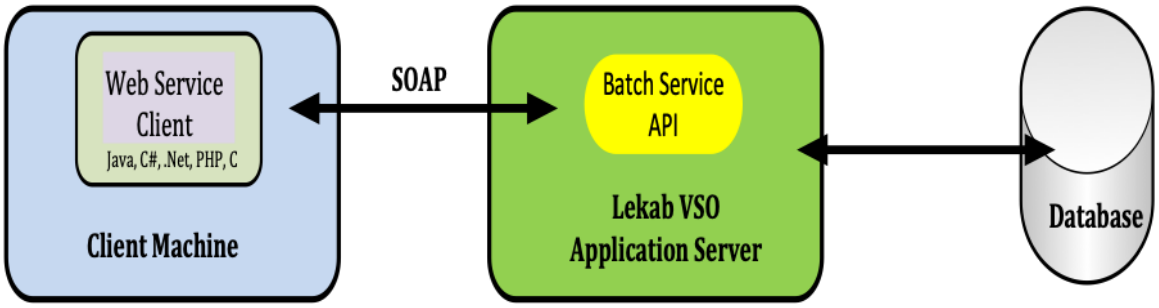

Figure 5. Web Service Client

For those not familiar with web services, Lekab will provide 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 Lekab.

1.4. Web Service Security

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`.

It is often important for the recipient to be able to determine the freshness of security semantics. In some cases, security semantics may be so stale that the recipient may decide to ignore it. The `<wsu:Timestamp>` element provides a mechanism for expressing the creation and expiration times of the security semantics in a message.

The elements that convey this information are:

- `<wsu:Created>2013-08-14T14:03:26Z</wsu:Created>`: Contains the time that the message was created.
- `<wsu:Expires>2013-08-14T14:13:26Z</wsu:Expires>`: Set by a sender or intermediary, this identifies when the message expires.

All times **MUST** be in the **UTC** time zone as specified by the XML Schema type (dateTime).

For connection security, the `HTTPS` shall be used to access the Lekab web services. The Lekab server certificate is signed by GlobalSign

In the following example in Listing 1, “WS-Security Header”, the username and password "**user1/user**" are used. The "Timestamp" is set to 10 minutes by default.

Listing 1, “WS-Security Header”

```
<soapenv:Header>
  <wsse:Security
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <wsu:Timestamp wsu:Id="TS-28">
      <wsu:Created>2013-08-14T14:03:26Z</wsu:Created>
      <wsu:Expires>2013-08-14T14:13:26Z</wsu:Expires>
    </wsu:Timestamp>
    <wsse:UsernameToken wsu:Id="UsernameToken-27">
      <wsse:Username>user1</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">user</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
```

1.5. Web Service Error Handling

Errors are received as SOAP faults in the SOAP Body. The SOAP fault contains the error details.

From the following example, in Listing 2, “SendBatchRequest” `SendBatchRequest` sends a `?` as recipient-which violates the message regulars, the server will reply a response with SOAP fault error codes and related description in Listing 3, “SendBatchRequest SOAP Fault”.

Listing 2, “SendBatchRequest”

```
<soapenv:Envelope xmlns:mes="http://www.lekab.com/schema/messaging/batch/v1/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" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
><wsu:Timestamp wsu:Id="TS-8"><wsu:Created>2013-09-19T12:13:00Z
</wsu:Created><wsu:Expires>2013-09-19T12:23:00Z</wsu:Expires>
</wsu:Timestamp><wsse:UsernameToken wsu:Id="UsernameToken-7"><wsse:Username>
user1</wsse:Username><wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">
user</wsse:Password></wsse:UsernameToken></wsse:Security></soapenv:Header>
   <soapenv:Body>
```

```
      <mes:SendBatchRequest>
         <mes:sender>Lekab</mes:sender>
          <mes:recipients>
            <!--1 or more repetitions:-->
            <mes:recipient>4670111111112</mes:recipient>
            <mes:recipient>?</mes:recipient>
         </mes:recipients>
         <!--Optional:-->
         <mes:message>SGVsbG8gd29ybGQ=LA=</mes:message>
      </mes:SendBatchRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Listing 3, “SendBatchRequest SOAP Fault”

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-3">
            <wsu:Created>2013-09-20T08:54:24.913Z</wsu:Created>
            <wsu:Expires>2013-09-20T08:59:24.913Z</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>
            <ns2:SendBatchFault xmlns:ns2=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
               <ns2:errorDetail>
                  <ns2:errorCode>14</ns2:errorCode>
                  <ns2:reasonCode>102</ns2:reasonCode>
                  <ns2:errorDescription>"?" is not a valid
recipient</ns2:errorDescription>
               </ns2:errorDetail>
            </ns2:SendBatchFault>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.6. Read Timeout Setting

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

to use a rather high read timeout.

1.7. How to Connect to Lekab Web Batch Services

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

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

[1] http://en.wikipedia.org/wiki/SOAP

Chapter 2. Batch Service API Operations

The Web Service Batch API has following available operations:

Batch Service Operation	Description
SendBatch	The SendBatch operation is used to send bulk SMS messages
BatchInfo	This operation is used for retrieving batch message status
BatchMessageId	This operation is used for retrieving message ids and the timestamp of the status
BatchMessageStatus	The get message status operation is used for querying the Lekab system for the message send status

2.1. "SendBatch" Operation

2.1.1. SendBatchRequest

The following parameters used for sending batch SMS messages.

//Element **/@attribute**	**Type**	**Mandatory/ Optional/ Unsupported**	**Default value**	**Max Length**	**Description**
//SendBatchRequest /sender	string	Mandatory		11/15	The sender id. Max length 11 if alphanumeric and 15 if an MSISDN or shortcode.

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//SendBatchRequest /recipients /recipient	string	Optional		15	The MSISDN of the recipient. It must be the recipient phone number including country code e.g. 46706352602. At least one recipient is required if no attachment file is used.
//SendBatchRequest /referenceId	string	Optional		150	The id is defined by the users to trace a sent message. I.e. a unique random reference id generated for a message sent to the recipient (4794170002) – "0a6a59483dd74c22b25d90fae7cdb2c6:4794170002"
//SendBatchRequest /attributes /name /value		Optional			Extra attributes are used for the request. Those attributes can be "Application" and its current "Version"

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//SendBatchRequest /message		Optional			The message to be sent. This field shall be used if no attachment file is used.
//SendBatchRequest /data /textDataFile		Optional			"textDataFile" is attachment file

The `SendBatchRequest` shall at least contain one source for the SMS data - "recipient" and "message" or "textDataFile".
The messages in both the SOAP message field and the attachment will be encoded automatically into Base64 before sending.
In the attachment "textDataFile", the following syntaxes can be used:

```
Syntax-1: Mobile number
Syntax-2: Mobile number, ReferenceId
Syntax-3: Mobile number, ReferenceId; Message text
```

All mobiles schemed with **Syntax-1** and **Syntax-2** will receive the same message, which is defined in the SOAP message field; all mobiles schemed with **Syntax-3** will receive different messages, which are defined in the "Message text" field in the attachment.

In Figure 6, “Attachment.txt”, the "Attachment.txt" has 6 mobiles. 3 mobiles are schemed with **Syntax-2** and 3 mobiles schemed with **Syntax-3**.

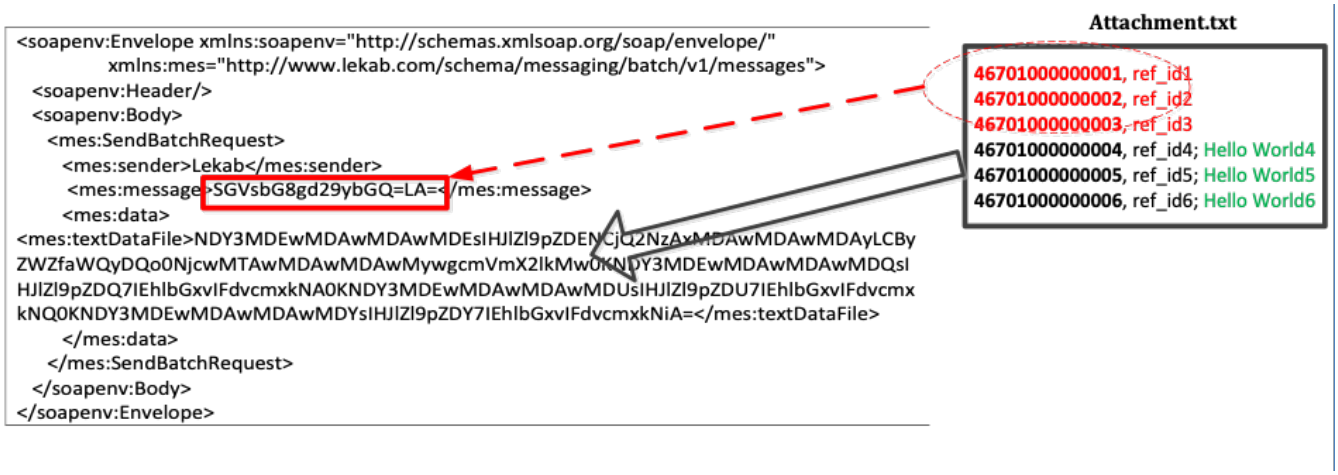

Figure 6. Attachment.txt

Both the message and the attachment are in Base64.

After sending SendBatchRequest, those mobiles ("4670100000001", "4670100000002" and "4670100000003") will receive the same message "Hello World" while other mobiles "4670100000004", "4670100000005" and "4670100000006" will receive message "Hello World4", "Hello World5" and "Hello World6" respectively.

The messages in both the SOAP message field and the attachment "textDataFile" shall be partly or entirely encoded with UTF-8 URL-encoding if the messages contains special characters, such as, line breaks "\ n", "+" and "%". Figure 7, “URL Encoding” shows how a message with special characters shall be handled before sending. The message can be partly encoded with URL-encoded as "scheme 1" (It is sufficient to encode those special characters with URL-encoding) or the message is entirely encoded with URL-encoding as "scheme 2". The "scheme 2" message is bigger than the "scheme 1" message, thus the "scheme 2" message will consume more resource than the "scheme 1".

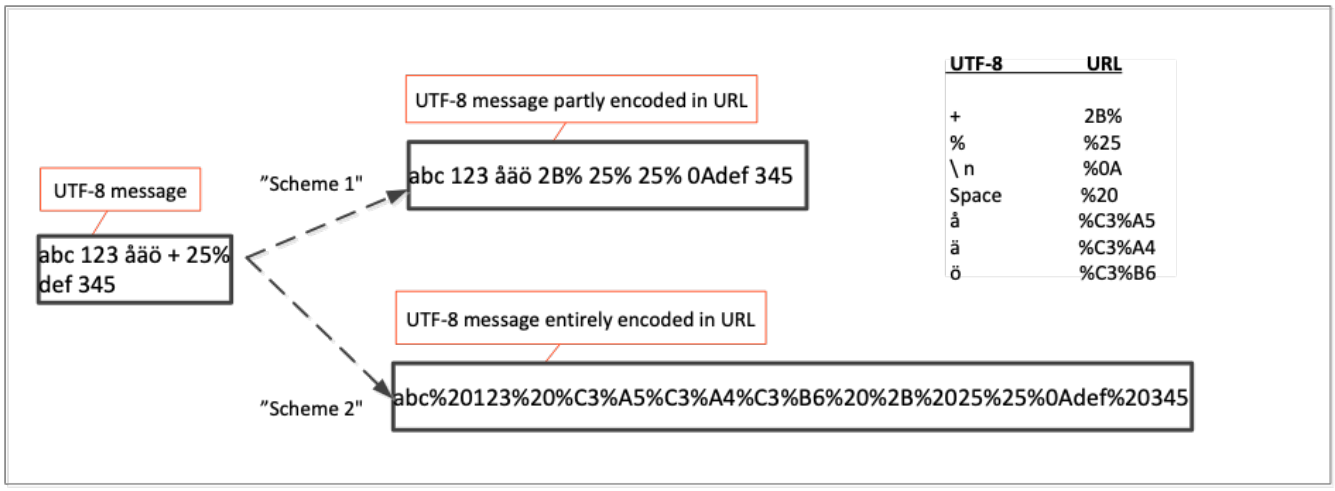

Figure 7. URL Encoding

A well defined SendBatchRequest template is shown in Listing 4, "Well defined `SendBatchRequest`".

Listing 4, "Well defined `SendBatchRequest`"

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:SendBatchRequest>
         <mes:sender>?</mes:sender>
         <!--Optional:-->
         <mes:referenceId>?</mes:referenceId>
         <!--Optional:-->
         <mes:recipients>
            <!--1 or more repetitions:-->
            <mes:recipient>?</mes:recipient>
         </mes:recipients>
         <!--Optional:-->
         <mes:message>cid:588656642218</mes:message>
         <!--Optional:-->
         <mes:data>
            <mes:textDataFile>cid:549771355682</mes:textDataFile>
         </mes:data>
```

```
         <!--Optional:-->
         <mes:attributes>
            <!--1 or more repetitions:-->
            <mes:attribute>
               <mes:name>?</mes:name>
               <mes:value>
                  <!--You have a CHOICE of the next 8 items at this level-->
                  <mes:string>?</mes:string>
                  <mes:boolean>?</mes:boolean>
                  <mes:integer>?</mes:integer>
                  <mes:long>?</mes:long>
                  <mes:float>?</mes:float>
                  <mes:double>?</mes:double>
                  <mes:base64Binary>cid:1074486748344</mes:base64Binary>
                  <mes:date>?</mes:date>
               </mes:value>
            </mes:attribute>
         </mes:attributes>
      </mes:SendBatchRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

2.1.2. SendBatchResponse

A list of messageStatus objects are returned.

//Element **/@attribute**	**Type**	**Max Length**	**Description**
//SendBatchResponse /messageStatus /statusCode	integer		The status code of the message
//SendBatchResponse /messageStatus /statusText	string	100	The status text for the given status code.
//SendBatchResponse /messageStatus /id	string	150	The message/batch id (use this message/batch id for the MessageStatus request)
//SendBatchResponse /messageStatus /attributes			Unsupported

2.2. "BatchInfo" Operation

2.2.1. BatchInfoRequest

//Element /@attribute	Type	Mandatory/ Optional/* Unsupported	Default value	Max Length	Description
//BatchInfoRequest /Id	string	Mandatory		150	The batch id is received when sending message
//BatchInfoRequest /attributes		Unsupported			This is used for special cases where extra attributes are used for the request. Not for normal use.

A well defined `BatchInfoRequest` template is shown in Listing 5, “BatchInfoRequest”.

Listing 5, “BatchInfoRequest”

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:BatchInfoRequest>
         <mes:id>?</mes:id>
         <!--Optional:-->
         <mes:attributes>
            <!--1 or more repetitions:-->
            <mes:attribute>
               <mes:name>?</mes:name>
               <mes:value>
                  <!--You have a CHOICE of the next 8 items at this level-->
                  <mes:string>?</mes:string>
                  <mes:boolean>?</mes:boolean>
                  <mes:integer>?</mes:integer>
                  <mes:long>?</mes:long>
                  <mes:float>?</mes:float>
                  <mes:double>?</mes:double>
                  <mes:base64Binary>cid:1204126485728</mes:base64Binary>
                  <mes:date>?</mes:date>
               </mes:value>
            </mes:attribute>
         </mes:attributes>
      </mes:BatchInfoRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

2.2.2. BatchInfoResponse

//Element /@attribute	Type	Max Length	Description
//BatchInfoResponse /messageStatus /statusCode			The status code of the message
//BatchInfoResponse /messageStatus /statusText	string	150	The status text for the given status code
//BatchInfoResponse /messageStatus /Id	dateTime		The message/batch id (use this message/batch id for the MessageStatus request)
//BatchInfoResponse /messageStatus /attributes	integer		Unsupported

2.3. "BatchMessageId" Operation

2.3.1. BatchMessageIdRequest

This operation is used to query the VSO system for a specific batch.

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//BatchMessage IdRequest /Id	string	Mandatory		150	The batch id is received when sending message
//BatchMessage IdRequest /attributes		Unsupported			This is used for special cases where extra attributes are used for the request. Not for normal use.

A well defined `BatchMessageIdRequest` template is shown in Listing 6, “BatchMessageIdRequest”.

Listing 6, “BatchMessageIdRequest”

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
```

```
    <soapenv:Header/>
    <soapenv:Body>
       <mes:BatchMessageIdRequest>
          <mes:id>?</mes:id>
          <!--Optional:-->
          <mes:attributes>
             <!--1 or more repetitions:-->
             <mes:attribute>
                <mes:name>?</mes:name>
                <mes:value>
                   <!--You have a CHOICE of the next 8 items at this level-->
                   <mes:string>?</mes:string>
                   <mes:boolean>?</mes:boolean>
                   <mes:integer>?</mes:integer>
                   <mes:long>?</mes:long>
                   <mes:float>?</mes:float>
                   <mes:double>?</mes:double>
                   <mes:base64Binary>cid:130668456487</mes:base64Binary>
                   <mes:date>?</mes:date>
                </mes:value>
             </mes:attribute>
          </mes:attributes>
       </mes:BatchMessageIdRequest>
    </soapenv:Body>
 </soapenv:Envelope>
```

2.3.2. BatchMessageIdResponse

//Element **/@attribute**	**Type**	**Max Length**	**Description**
//BatchMessageIdResponse /attributes			Unsupported
//BatchMessageIdResponse /messageId	string	150	A list of message ids received
//BatchMessageIdResponse /timeStamp	dateTime		The time the message was received

2.4. "BatchMessageStatus" Operation

2.4.1. BatchMessageStatusRequest

This operation is used to query the VSO system for the batch send status.

//Element /@attribute	Type	Mandatory/ Optional/ Unsupported	Default value	Max Length	Description
//BatchMessageStatusRequest /@markStatusesRead	Boolean	Mandatory	false		Mark the retrieved messages as read
//BatchMessageStatusRequest /@maxNumberOfStatuses	integer	Mandatory	1000		Number of message statuses to retrieve (not applicable if message ids specified)
//BatchMessageStatusRequest /batchId		Optional			The batch id is received in "SendBatchResponse"
//BatchMessageStatusRequest /batchReferenceId /batchMessageReferenceIds		Optional			The client defines this id as reference id when sending message
//BatchMessageStatusRequest /messageIds /messageId	string	Optional		150	A list of message ids received from "BatchMessageId" operation
//BatchtMessageStatusRequest /attributes		Unsupported			This is used for special cases where extra attributes are used for the request. Not for normal use.

The `BatchMessageStatusRequest` shall at least contain one element – `batchId` or `batchReferenceId` or `messageId`.
Without providing any ids in the `BatchMessageStatusRequest`, the latest statuses from all batches will be received.

A well defined `BatchMessageStatusRequest` template is shown in Listing 7, “BatchMessageStatusRequest”.

Listing 7, “BatchMessageStatusRequest”

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:BatchMessageStatusRequest markStatusesRead="false" maxNumberOfStatuses=
"1000">
         <!--Optional:-->
         <mes:batchId>?</mes:batchId>
         <!--Optional:-->
         <mes:batchReferenceId>?</mes:batchReferenceId>
         <!--Optional:-->
         <mes:messageIds>
            <!--1 or more repetitions:-->
            <mes:messageId>?</mes:messageId>
         </mes:messageIds>
         <!--Optional:-->
         <mes:batchMessageReferenceIds>
            <!--1 or more repetitions:-->
            <mes:messageId>?</mes:messageId>
         </mes:batchMessageReferenceIds>
         <!--Optional:-->
         <mes:attributes>
            <!--1 or more repetitions:-->
            <mes:attribute>
               <mes:name>?</mes:name>
               <mes:value>
                  <!--You have a CHOICE of the next 8 items at this level-->
                  <mes:string>?</mes:string>
                  <mes:boolean>?</mes:boolean>
                  <mes:integer>?</mes:integer>
                  <mes:long>?</mes:long>
                  <mes:float>?</mes:float>
                  <mes:double>?</mes:double>
                  <mes:base64Binary>cid:490833012216</mes:base64Binary>
                  <mes:date>?</mes:date>
               </mes:value>
            </mes:attribute>
         </mes:attributes>
      </mes:BatchMessageStatusRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

2.4.2. BatchMessageStatusResponse

//Element /@attribute	Type	Max Length	Description
//BatchMessageStatusResponse /messageStatus /statusCode	integer		The status code of the message
//BatchMessageStatusResponse /messageStatus /statusText	string	100	The status text for the given status code.
//BatchMessageStatusResponse /messageStatus /id	string	150	The batch id for the MessageStatus request
//BatchMessageStatusResponse /messageStatus /sender	string	15	The sender of the message
//BatchMessageStatusResponse /messageStatus /recipient	string	15	The recipient of the message
//BatchMessageStatusResponse /messageStatus /time	dateTime		The timestamp of the status
//BatchMessageStatusResponse /messageStatus /attributes			Extra attributes describing the messages such as "markStatusesRead" and "maxNumberOfStatuses"
//BatchMessageStatusResponse /attributes			Unsupported

2.5. Message Status and Fault Handling

The message sent status can be received in code from the corresponded response message. When there is an error, a reason code and error detail will be received in the SOAP fault message.

The errors in batch service operations will be handled by SOAP fault functions in the following table:

Batch Service Operation	SOAP Fault Function
SendBatch	SendBatchFault
BatchInfo	BatchInfoFault
BatchMessageId	BatchMessageIdFault
BatchMessageStatus	BatchMessageStatusFault

2.5.1. Status Code

Status codes are defined for different message statuses. E.g. status code "0" means no errors for a message from "received" to "processing" and "validating" in the VSO.

Status Code	Status Text	Description
0	OK	Ok
1	RECEIVED	Received
2	PROCESSING	Processing
3	VALIDATING	Validating
10	ERROR	Unexpected error
11	ERROR_QUOTA	Quota exceeded
12	ERROR_BATCH_SIZE	Maximum batch size exceeded
13	ACCESS_ERROR	Access Denied
14	VALIDATION_ERROR	Validation error
15	ERROR_SEND_TIME	Dropped due to send time restrictions

2.5.2. Reason Code

Reason codes are defined to differentiate errors. The following reason codes can be received in the SOAP fault when sending messages.

Reason Code	Error Text	Description
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 exceeded (Only returned if the account has an attached monthly quota)
12	BATCH_SIZE_EXCEEDED	The batch size exceeds the allowed

Reason Code	Error Text	Description
101	SENDER_INVALID	"\{}" is not a valid sender. \{} is replaced with the sender
101	SENDER_REQUIRED	Sender is required for this message
102	RECIPIENT_INVALID	"\{}" is not a valid recipient. \{} is replaced with the invalid recipient.
102	RECIPIENT_REQUIRED	At least one recipient is required
103	INVALID_MESSAGE_LENGTH	The length of the SMS message is invalid
103	BATCH_DATA_REQUIRED	Data must be provided either in an attachment or in the SOAP request
104	SCHEDULED_DELIVERY_INVALID	The scheduled delivery time is not a valid date or is not in the future.
105	VALID_TO_INVALID	The validity period is not a valid date or is not in the future.
106	INVALID_REFERENCE_ID_LENGTH	Reference ID can’t be longer than MAX_REFERENCE_ID_LENGTH=50
107	INVALID_NUMBER_OF_STATUSES	Invalid number of statuses requested. MAX_STATUSES=10000
110	ID_REQUIRED	Id required
110	INVALID_ID	The reference id specified is not found
111	NO_BATCH_FOUND_FOR_ID	No batch found for id \"\{}\"

Chapter 3. Appendices

3.1. Batch Service Operation Examples

The examples are in SOAP format only. For code examples in *Java, C# .Net, PHP or C* etc. contact **Lekab Communication Systems**. The SOAP examples are formatted for readability and made with SoapUI http://www.soapui.org/.

3.1.1. Create a New SoapUI Project

A "BatchWSapi_Example" SoapUI is created and the Batch API http://localhost:8080/batch/messaging-batch-v1.wsdl is used as in Figure 8, “Create new project”.

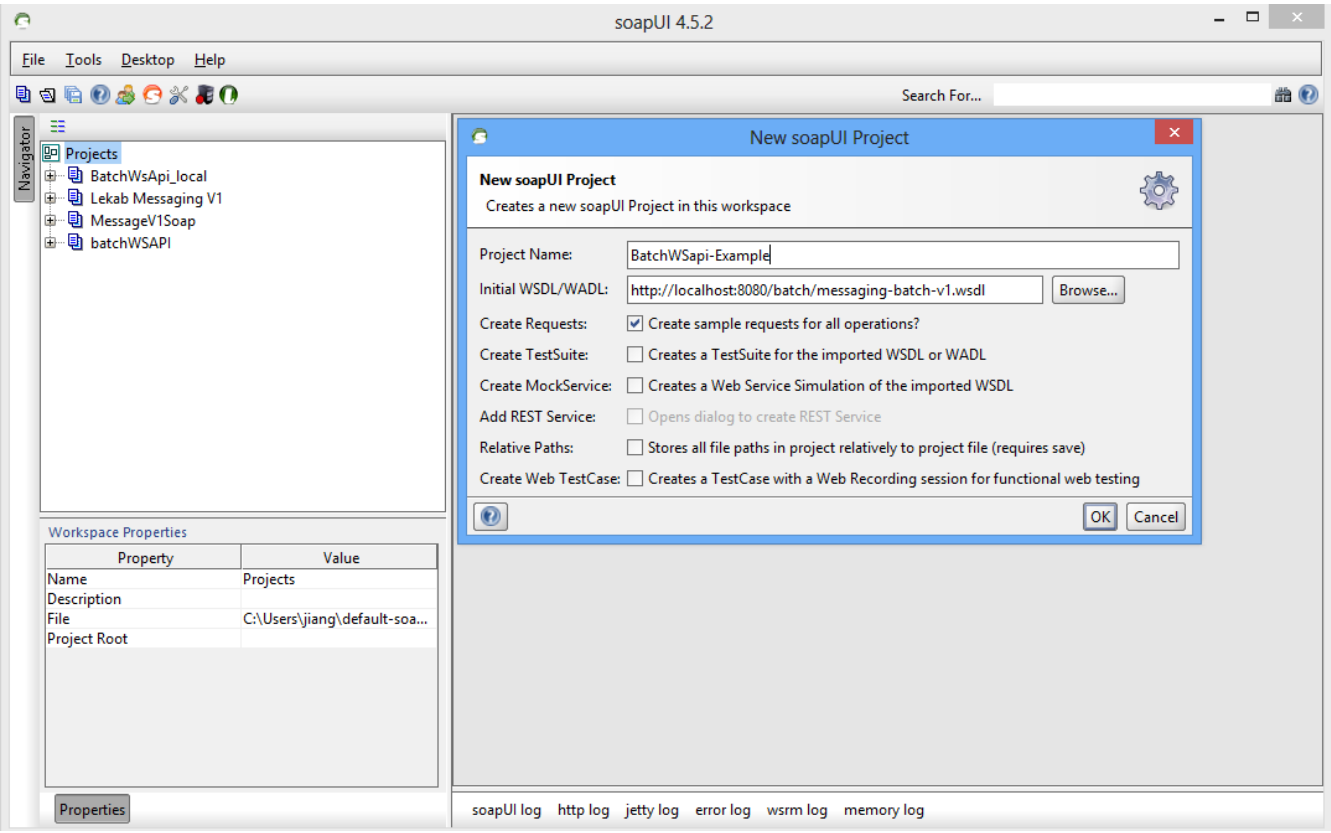

Figure 8. Create new project

Configure WS-Security

Add and configure an outgoing WS-security "BatchWSconfig1" in Figure 9, “Configure outgoing WS-security”.

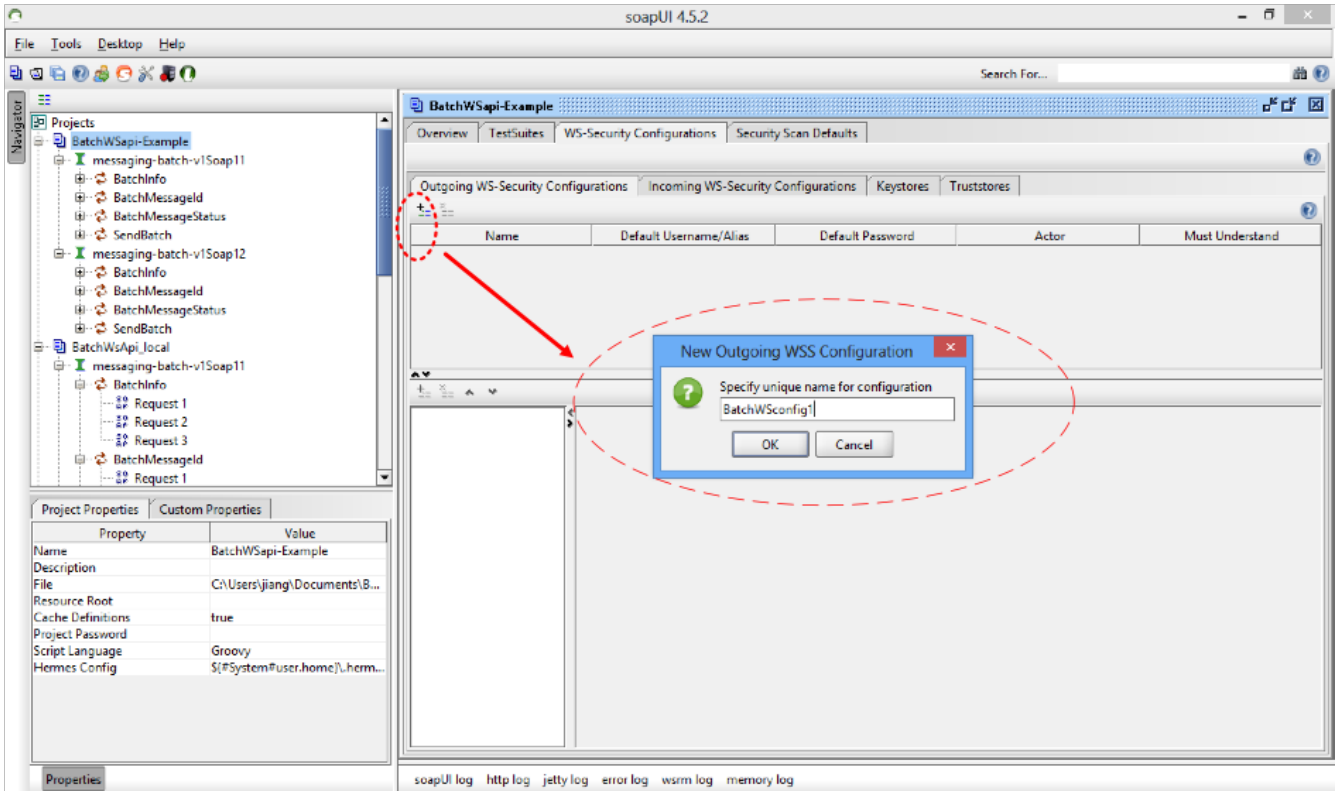

Figure 9. Configure outgoing WS-security

Add WSS Entries to "BatchWSconfig1"

Add Username in Figure 10, “Configure username and password”:

Username/Password – "user1/user"

Password Type – PasswordText

Adds a created – checked (after the Username is created, the "Adds a created" shall be "unchecked")

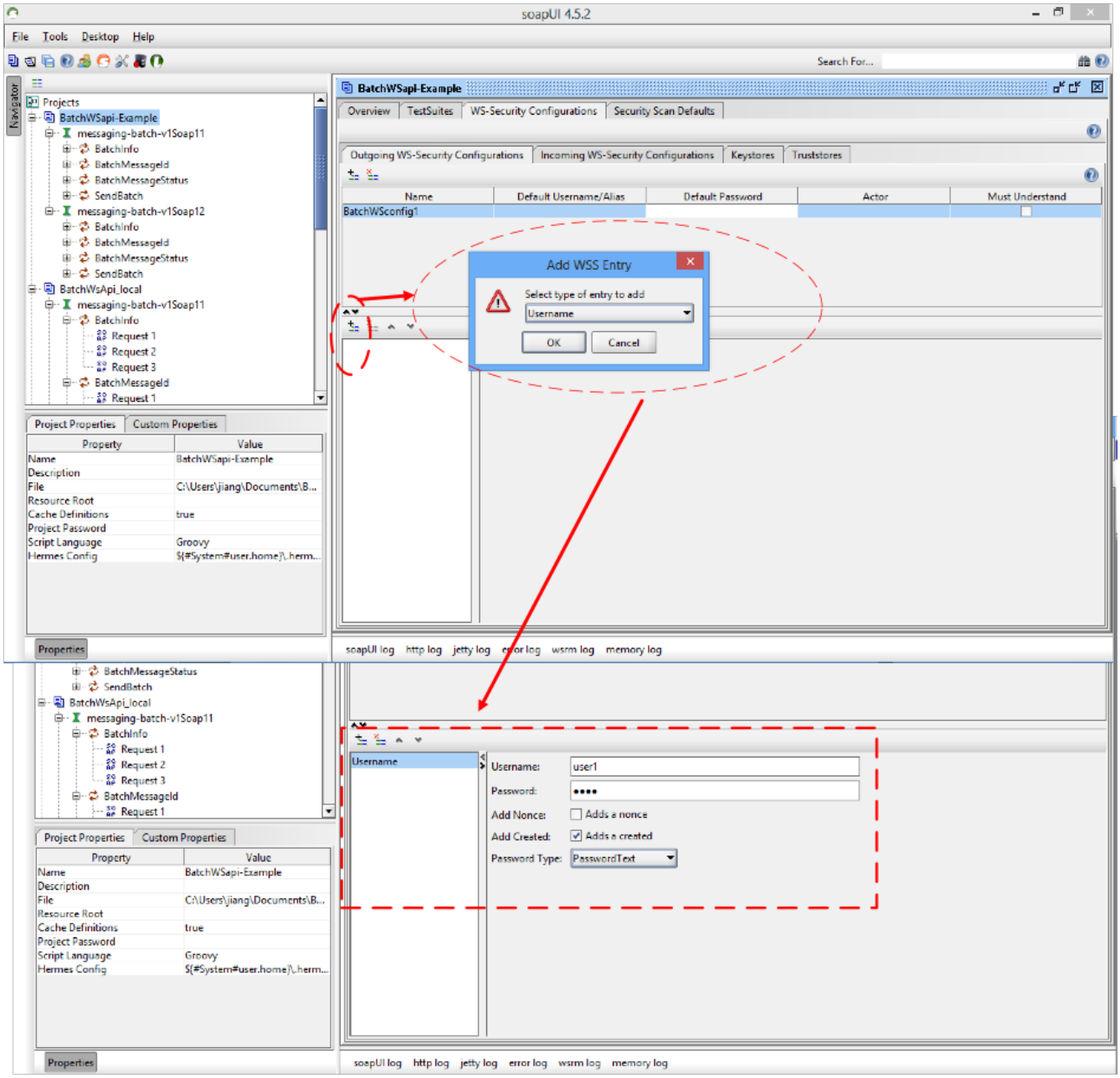

Figure 10. Configure username and password

Add Timestamp

To add a timestamp is the same way as to add a username in Figure 11, “Uncheck millisecond precision”:

Time To Live - 600

Millisecond Precision – unchecked

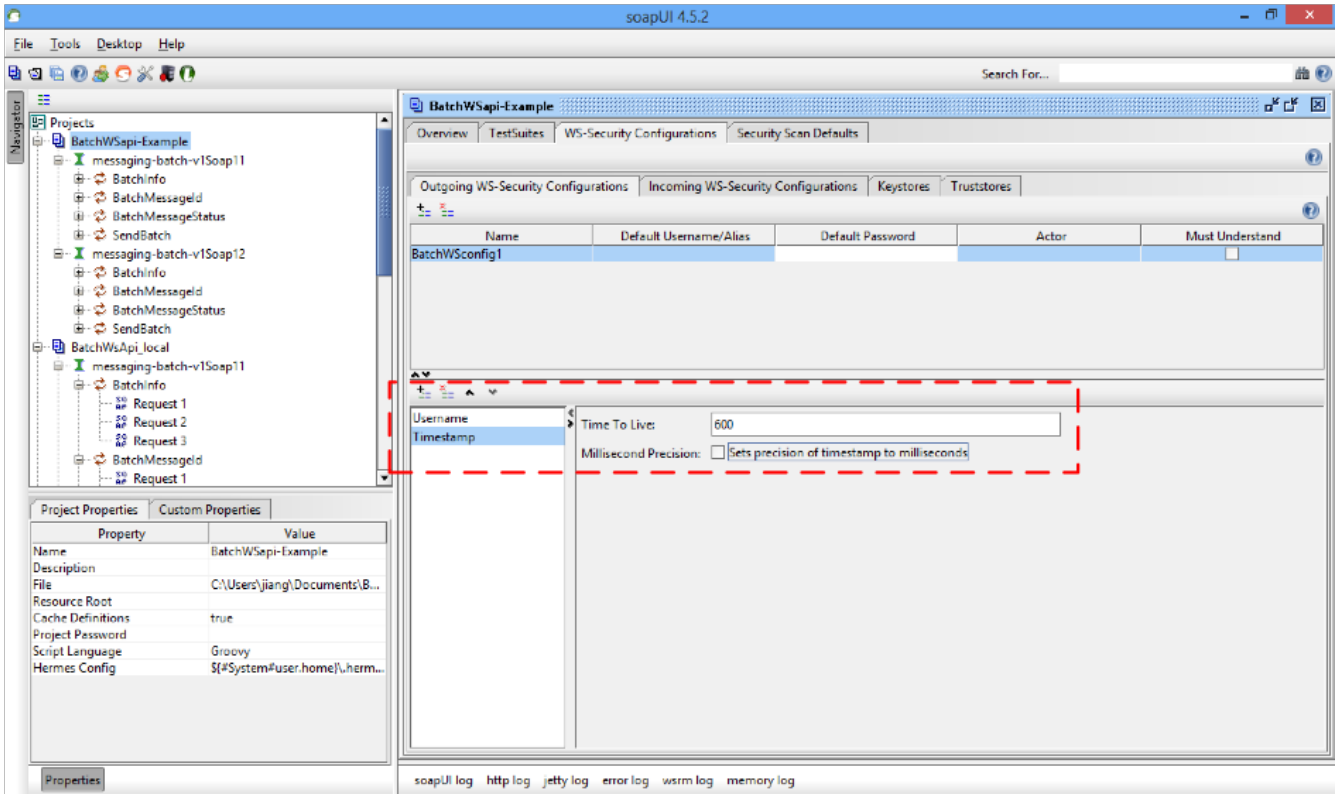

Figure 11. Uncheck millisecond precision

The created WS configuration "BatchWSconfig1" shall be selected for validation to each new created "Request" as in Figure 12, “BatchWSconfig1”.

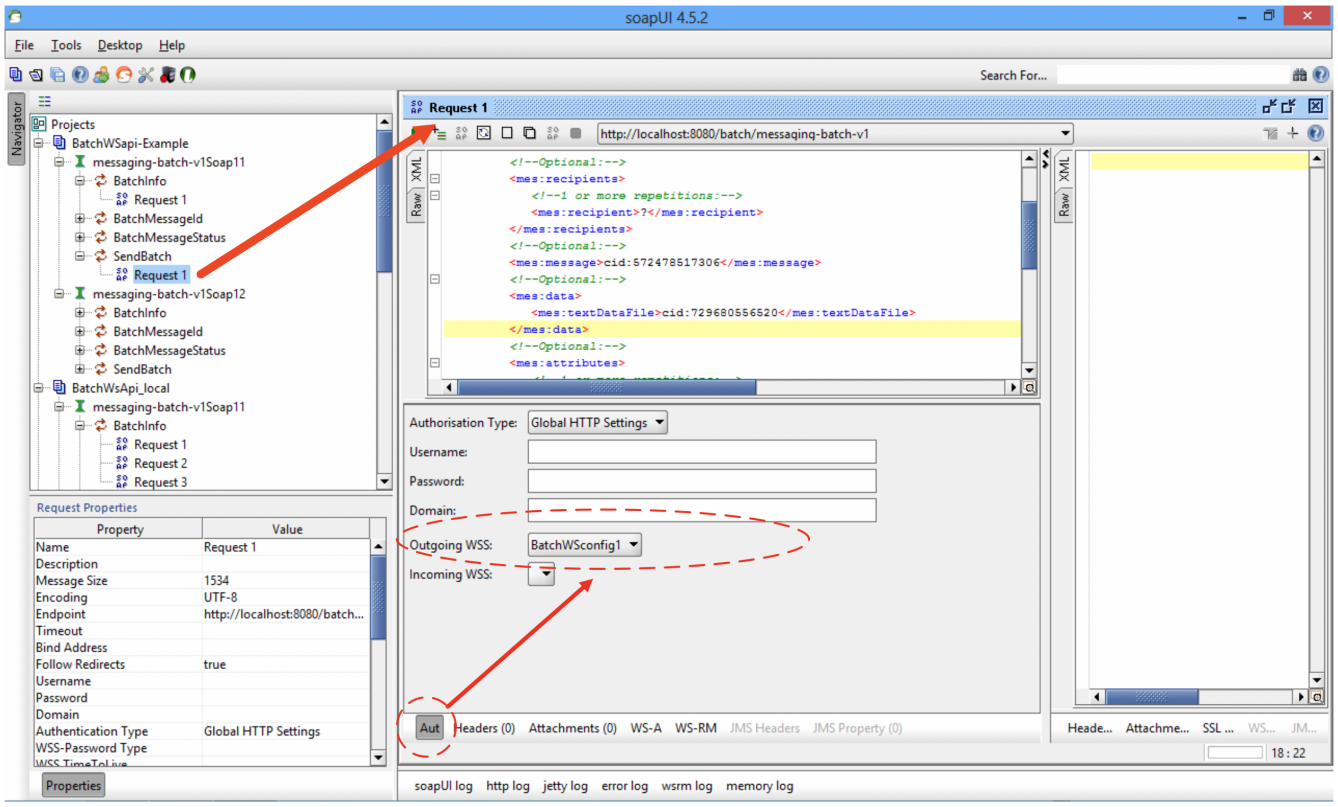

Figure 12. BatchWSconfig1

SOAP Request Body

The following parameters are used in the SendBatchRequest body:

```
Sender  "Lekab"
Recipients  "4670111111110"
ReferenceId  "Ref00"
SOAP message field  "Hello world" is encoded in Base64 "SGVsbG8gd29ybGQ=LA="
Attributes: Application  Lekab VSO Link; Version  1.0.3.0
```

The following syntaxes and messages in UTF-8 are used in the attached "textDataFile":

UTF-8 Text

```
46701000000001
46701000000002
46701000000003,ref_id03
46701000000004,ref_id04
46701000000005,ref_id05;Hello World5+Best Wishes 100%
                        Yours sincerely
46701000000006,ref_id06;Hello World6 + Best Wishes 100%
                        Kind
                        regards
46701000000007,ref_id07;Hello World7+Best Wishes 100%
                        Best regards
46701000000008,ref_id08;Hello World8 + Best Wishes 100%
                        Yours
                        faithfully
```

The Message texts in the attachment shall be encoded in URL before sending because the Message texts contain special characters. The Message texts sent to the mobiles "46701000000005" and "46701000000006" are partly encoded in URL; and the Message texts sent to the mobiles "46701000000007" and "46701000000008" are entirely encoded in URL:

URL Encoding

```
46701000000001
46701000000002
46701000000003,ref_id03
46701000000004,ref_id04
46701000000005,ref_id05;Hello World52B%Best Wishes 100%25%0AYours sincerely
46701000000006,ref_id06;Hello World6 2B% Best Wishes 100%25%0AKind%0Aregards
46701000000007,ref_id07;Hello%20World7%2BBest%20Wishes%20100%25%0ABest%20regards
46701000000008,ref_id08;Hello%20World8%2BBest%20Wishes%20100%25%0AYours%0Afaithfully
```

The URL encoded attachment is saved in a text file "SoapBatchAttachemntURL20130902.txt". This attachment file can be inserted into the Soap Request "textDataFile" field by right click on the mouse between <mes:textDataFile> and </mes:textDataFile>. The "Insert file as Base64" is selected to insert the attachment file as shown in Figure 13, “Insert file as Base64” and Figure 14, “Choose file”.

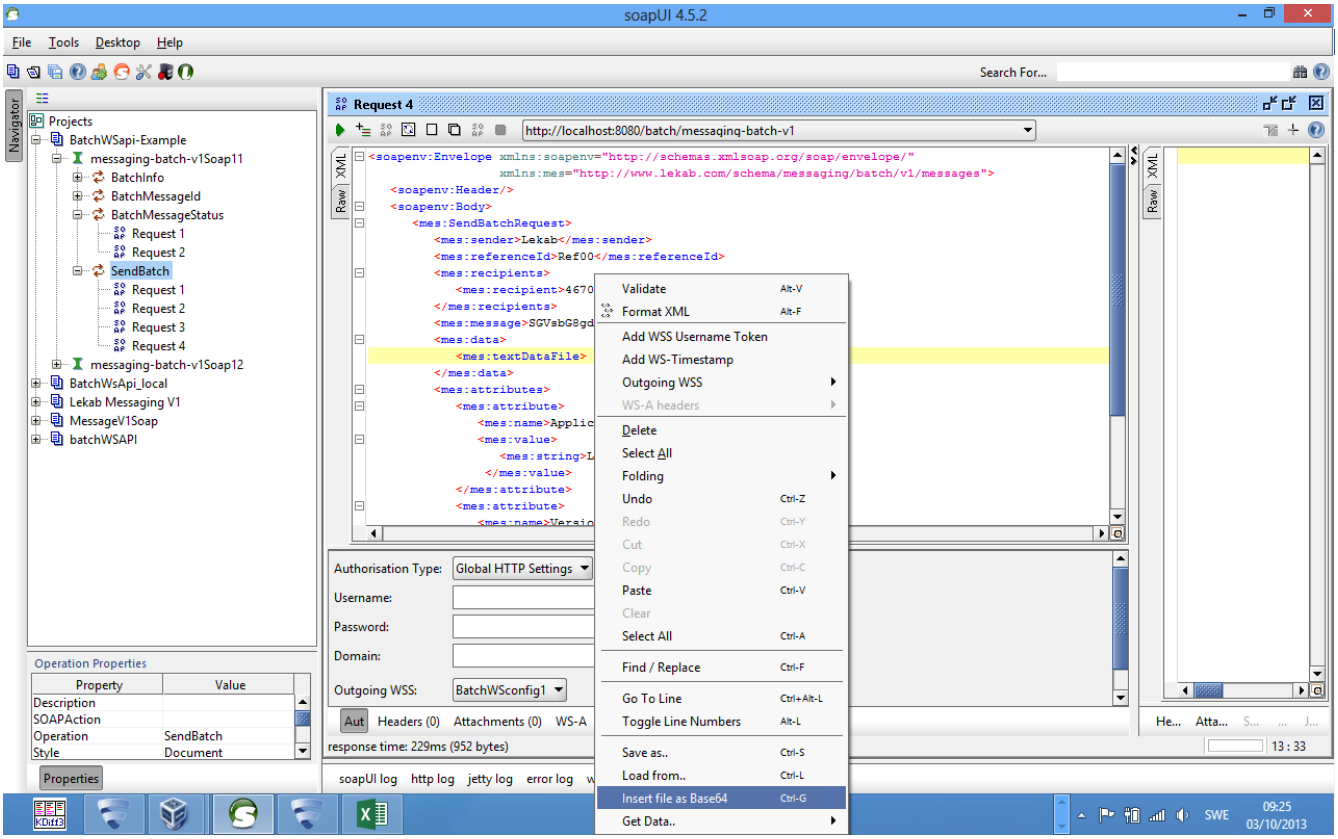

Figure 13. Insert file as Base64

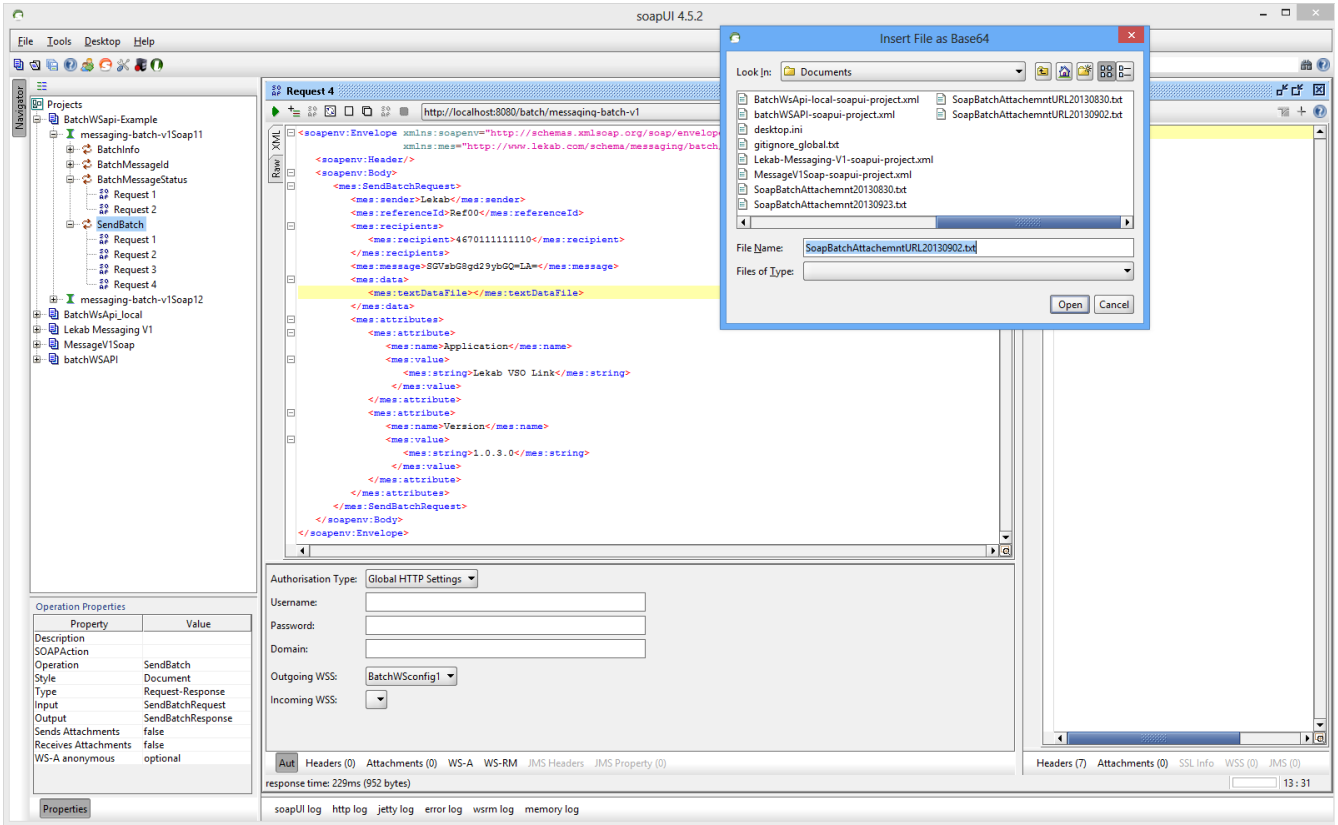

Figure 14. Choose file

3.1.2. SendBatch

SendBatchRequest


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:mes="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:SendBatchRequest>
         <mes:sender>Lekab</mes:sender>
         <mes:referenceId>Ref00</mes:referenceId>
         <mes:recipients>
            <mes:recipient>4670111111110</mes:recipient>
         </mes:recipients>
         <mes:message>SGVsbG8gd29ybGQ=LA=</mes:message>
         <mes:data>
<mes:textDataFile>NDY3MDEwMDAwMDAwMDENCjQ2NzAxMDAwMDAwMDAyDQo0NjcwMTAwMDAwMDAwMywgcmVm
X2lkMDMNCjQ2NzAxMDAwMDAwMDA0LCByZWZfaWQwNA0KNDY3MDEwMDAwMDAwMDUsIHJlZl9pZDA1O0hlbGxvIF
dvcmxkNSUyQkJlc3QgV2lzaGVzIDEwMCUyNSUwQVlvdXJzIHNpbmNlcmVseQ0KNDY3MDEwMDAwMDAwMDYsIHJl
Zl9pZDA2O0hlbGxvIFdvcmxkNiAlMkIgQmVzdCBXaXNoZXMgMTAwJTI1JTBBS2luZCUwQXJlZ2FyZHMNCjQ2Nz
AxMDAwMDAwMDA3LCByZWZfaWQwNztIZWxsbyUyMFdvcmxkNyUyQkJlc3QlMjBXaXNoZXMlMjAxMDAlMjUlMEFC
ZXN0JTIwcmVnYXJkcw0KNDY3MDEwMDAwMDAwMDgsIHJlZl9pZDA4O0hlbGxvJTIwV29ybGQ4JTIwJTJCJTIwQm
VzdCUyMFdpc2hlcyUyMDEwMCUyNSUwQVlvdXJzJTBBZmFpdGhmdWxseQ0K</mes:textDataFile>
         </mes:data>
         <mes:attributes>
            <mes:attribute>
               <mes:name>Application</mes:name>
               <mes:value>
                  <mes:string>Lekab VSO Link</mes:string>
                </mes:value>
            </mes:attribute>
            <mes:attribute>
               <mes:name>Version</mes:name>
               <mes:value>
                  <mes:string>1.0.3.0</mes:string>
                </mes:value>
            </mes:attribute>
         </mes:attributes>
      </mes:SendBatchRequest>
   </soapenv:Body>
</soapenv:Envelope>
```


Both "Hello World" and the entire attachment are encoded in Base64.

SendBatchResponse


```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
         <wsu:Timestamp wsu:Id="TS-4">
            <wsu:Created>2013-10-03T07:41:33.780Z</wsu:Created>
            <wsu:Expires>2013-10-03T07:46:33.780Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <ns2:SendBatchResponse xmlns:ns2=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
         <ns2:statusCode>1</ns2:statusCode>
         <ns2:statusText>Received</ns2:statusText>
         <ns2:id>22db3135074d41c4a5d78438036c38bf</ns2:id>
      </ns2:SendBatchResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.3. BatchInfo

BatchInfoRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:BatchInfoRequest>
         <mes:id>22db3135074d41c4a5d78438036c38bf</mes:id>
      </mes:BatchInfoRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

BatchInfoResponse

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-5">
            <wsu:Created>2013-10-03T07:43:32.157Z</wsu:Created>
            <wsu:Expires>2013-10-03T07:48:32.157Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <ns2:BatchInfoResponse xmlns:ns2=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
         <ns2:statusCode>0</ns2:statusCode>
         <ns2:statusText>Ok</ns2:statusText>
```

```
         <ns2:id>22db3135074d41c4a5d78438036c38bf</ns2:id>
      </ns2:BatchInfoResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.4. BatchMessageId

BatchMessageIdRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:BatchMessageIdRequest>
         <mes:id>22db3135074d41c4a5d78438036c38bf</mes:id>
      </mes:BatchMessageIdRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

BatchMessageIdResponse

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-7">
            <wsu:Created>2013-10-03T07:48:07.793Z</wsu:Created>
            <wsu:Expires>2013-10-03T07:53:07.793Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <ns2:BatchMessageIdResponse xmlns:ns2=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
         <ns2:messageId>1-44</ns2:messageId>
         <ns2:messageId>1-42</ns2:messageId>
         <ns2:messageId>1-36</ns2:messageId>
         <ns2:messageId>1-38</ns2:messageId>
         <ns2:messageId>1-40</ns2:messageId>
         <ns2:messageId>1-39</ns2:messageId>
         <ns2:messageId>1-37</ns2:messageId>
         <ns2:messageId>1-41</ns2:messageId>
         <ns2:messageId>1-43</ns2:messageId>
      </ns2:BatchMessageIdResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.5. BatchMessageStatus

BatchMessageStatusRequest

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mes
="http://www.lekab.com/schema/messaging/batch/v1/messages">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:BatchMessageStatusRequest markStatusesRead="false" maxNumberOfStatuses=
"1000">
         <mes:batchId>22db3135074d41c4a5d78438036c38bf</mes:batchId>
      </mes:BatchMessageStatusRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

BatchMessageStatusResponse

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
         <wsu:Timestamp wsu:Id="TS-8">
            <wsu:Created>2013-10-03T07:49:56.219Z</wsu:Created>
            <wsu:Expires>2013-10-03T07:54:56.219Z</wsu:Expires>
         </wsu:Timestamp>
      </wsse:Security>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <ns2:BatchMessageStatusResponse xmlns:ns2=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
         <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:id>1-36</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>4670111111110</ns2:recipient>
            <ns2:time>2013-10-03T09:41:45.000+02:00</ns2:time>
         </ns2:batchMessageStatus>
         <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:id>1-37</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000002</ns2:recipient>
```

```
            <ns2:time>2013-10-03T09:41:47.000+02:00</ns2:time>
        </ns2:batchMessageStatus>
        <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:id>1-38</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000001</ns2:recipient>
            <ns2:time>2013-10-03T09:41:54.000+02:00</ns2:time>
        </ns2:batchMessageStatus>
        <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id04</ns2:batchMessageReferenceId>
            <ns2:id>1-39</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000004</ns2:recipient>
            <ns2:time>2013-10-03T09:41:51.000+02:00</ns2:time>
        </ns2:batchMessageStatus>
        <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id03</ns2:batchMessageReferenceId>
            <ns2:id>1-40</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000003</ns2:recipient>
            <ns2:time>2013-10-03T09:41:44.000+02:00</ns2:time>
        </ns2:batchMessageStatus>
        <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id06</ns2:batchMessageReferenceId>
            <ns2:id>1-41</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000006</ns2:recipient>
            <ns2:time>2013-10-03T09:41:43.000+02:00</ns2:time>
        </ns2:batchMessageStatus>
        <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id08</ns2:batchMessageReferenceId>
```

```
            <ns2:id>1-42</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000008</ns2:recipient>
            <ns2:time>2013-10-03T09:41:45.000+02:00</ns2:time>
         </ns2:batchMessageStatus>
         <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id05</ns2:batchMessageReferenceId>
            <ns2:id>1-43</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000005</ns2:recipient>
            <ns2:time>2013-10-03T09:41:47.000+02:00</ns2:time>
         </ns2:batchMessageStatus>
         <ns2:batchMessageStatus>
            <ns2:statusCode>2</ns2:statusCode>
            <ns2:statusText>DELIVERED</ns2:statusText>
            <ns2:batchId>22db3135074d41c4a5d78438036c38bf</ns2:batchId>
            <ns2:batchReferenceId>Ref00</ns2:batchReferenceId>
            <ns2:batchMessageReferenceId>ref_id07</ns2:batchMessageReferenceId>
            <ns2:id>1-44</ns2:id>
            <ns2:sender>Lekab</ns2:sender>
            <ns2:recipient>46701000000007</ns2:recipient>
            <ns2:time>2013-10-03T09:41:45.000+02:00</ns2:time>
         </ns2:batchMessageStatus>
      </ns2:BatchMessageStatusResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


3.2. URL Encoding Tool

A web based URL encoding tool (http://meyerweb.com/eric/tools/dencoder/) can be used to encode UTF-8 text to URL codes as shown in the Figure 15, “URL Encoder/Decoder”.

```
Hello World8+Best Wishes 100%
Yours
faithfully
```

The UTF-8 text:

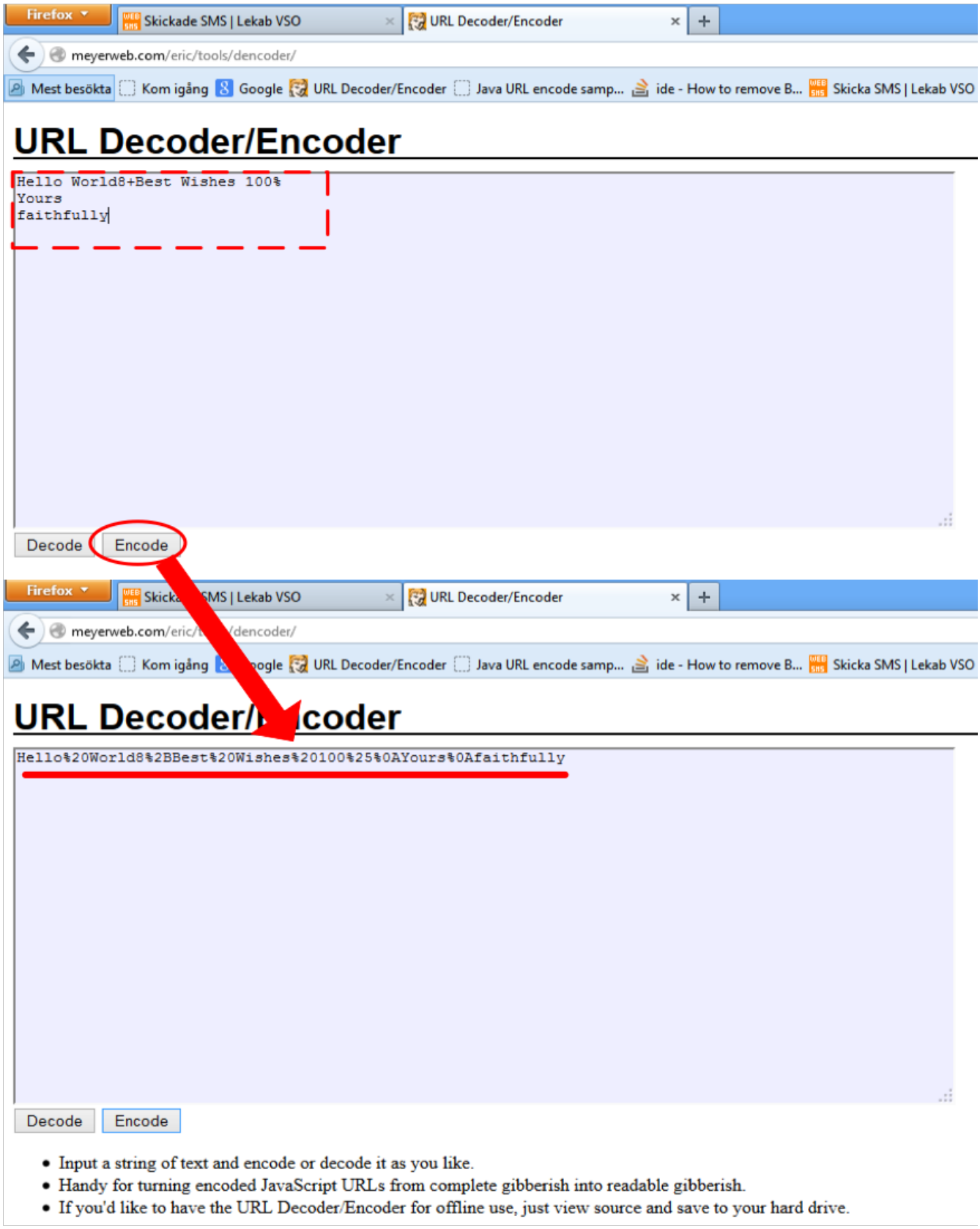

Figure 15. URL Encoder/Decoder

3.3. Base64 Encoding Tool

A web based Base64 encoding tool (http://www.motobit.com/util/base64-decoder-encoder.asp) can be used to encode UTF-8 text to Base64 as shown in the Figure 16, “Base64 Encoder/Decoder”.

The UTF-8 text: Hello World

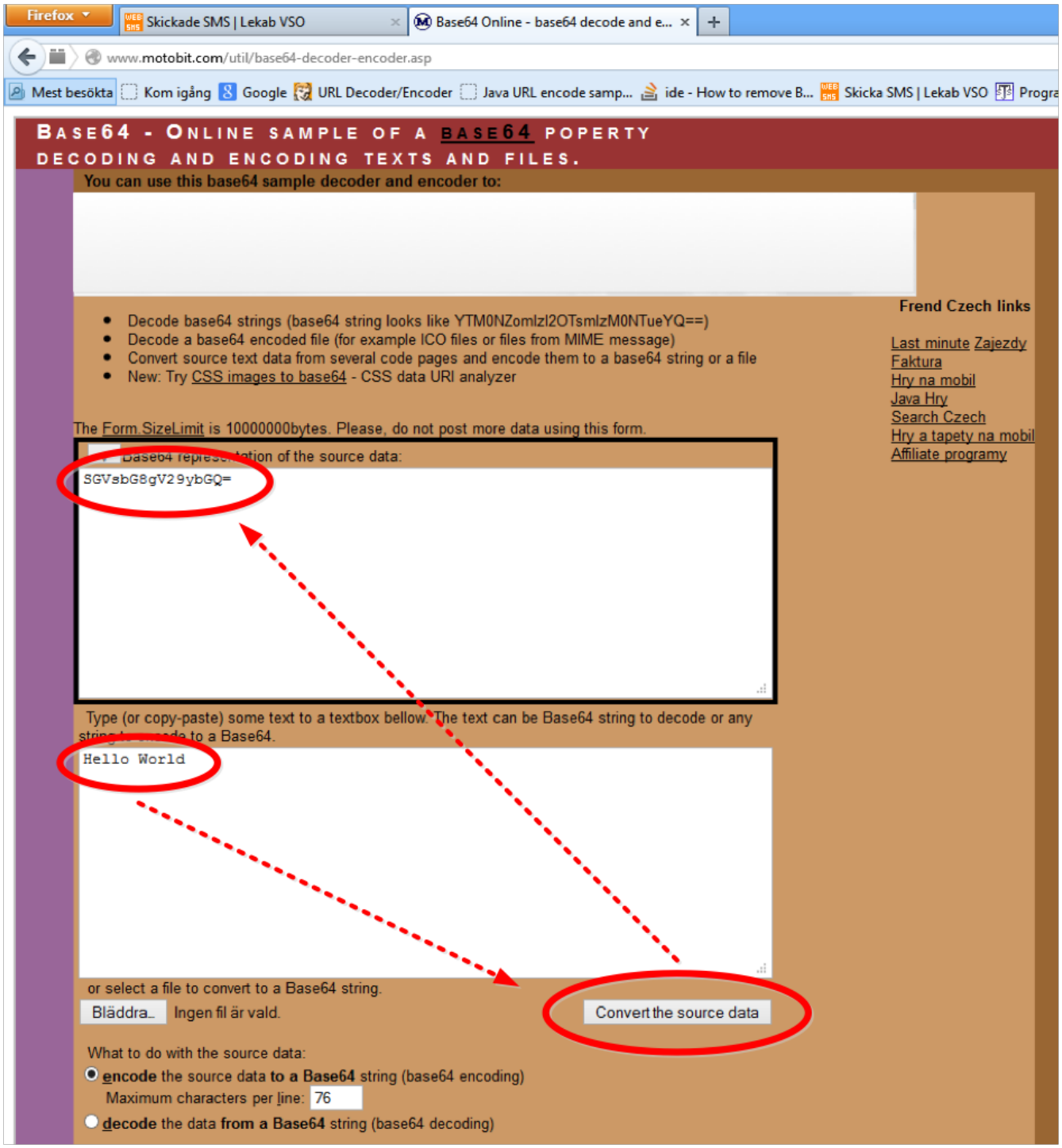

Figure 16. Base64 Encoder/Decoder

3.4. Lekab Web Batch Service WSDL

```
<wsdl:definitions targetNamespace=
"http://www.lekab.com/schema/messaging/batch/v1/messages" xmlns:wsdl=
"http://schemas.xmlsoap.org/wsdl/" xmlns:sch=
"http://www.lekab.com/schema/messaging/batch/v1/messages" xmlns:soap=
"http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12=
"http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns=
"http://www.lekab.com/schema/messaging/batch/v1/messages">
   <wsdl:types>
      <xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
```

```
jaxb:extensionBindingPrefixes="xjc" jaxb:version="2.0" targetNamespace=
"http://www.lekab.com/schema/messaging/batch/v1/messages" xmlns=
"http://www.lekab.com/schema/messaging/batch/v1/messages" xmlns:jaxb=
"http://java.sun.com/xml/ns/jaxb" xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema">
         <xsd:annotation>
            <xsd:appinfo>
               <jaxb:globalBindings>
                  <xjc:javaType adapter="com.lekab.jaxb2.util.DateTimeXmlAdapter"
name="org.joda.time.DateTime" xmlType="xsd:dateTime"/>
               </jaxb:globalBindings>
            </xsd:appinfo>
         </xsd:annotation>
         <xsd:element name="SendBatchRequest">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="sender" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="referenceId" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="recipients" type="Recipients"/>
                  <xsd:element minOccurs="0" name="message" type="xsd:base64Binary"/>
                  <xsd:element minOccurs="0" name="data" type="BatchData"/>
                  <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="SendBatchResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="statusCode" type="xsd:int"/>
                  <xsd:element minOccurs="0" name="statusText" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="id" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="SendBatchFault">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:sequence>
                     <xsd:element maxOccurs="unbounded" name="errorDetail" type=
"ErrorDetail"/>
                  </xsd:sequence>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="BatchInfoRequest">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="id" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
```

```
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchInfoResponse">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="statusCode" type="xsd:int"/>
               <xsd:element minOccurs="0" name="statusText" type="xsd:string"/>
               <xsd:element minOccurs="0" name="id" type="xsd:string"/>
               <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchInfoFault">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:sequence>
                  <xsd:element maxOccurs="unbounded" name="errorDetail" type=
"ErrorDetail"/>
               </xsd:sequence>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchMessageIdRequest">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="id" type="xsd:string"/>
               <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchMessageIdResponse">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element maxOccurs="unbounded" minOccurs="0" name="messageId"
type="xsd:string"/>
               <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchMessageIdFault">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:sequence>
                  <xsd:element maxOccurs="unbounded" name="errorDetail" type=
"ErrorDetail"/>
               </xsd:sequence>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:element>
      <xsd:element name="BatchMessageStatusRequest">
```

```
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element minOccurs="0" name="batchId" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="batchReferenceId" type="xsd:string
"/>
                  <xsd:element minOccurs="0" name="messageIds" type="MessageIds"/>
                  <xsd:element minOccurs="0" name="batchMessageReferenceIds" type=
"MessageIds"/>
                  <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
               </xsd:sequence>
               <xsd:attribute default="false" name="markStatusesRead" type=
"xsd:boolean"/>
               <xsd:attribute default="1000" name="maxNumberOfStatuses" type="xsd:int
"/>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="BatchMessageStatusResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element maxOccurs="unbounded" minOccurs="0" name=
"batchMessageStatus" type="BatchMessageStatus"/>
                  <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="BatchMessageStatusFault">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:sequence>
                     <xsd:element maxOccurs="unbounded" name="errorDetail" type=
"ErrorDetail"/>
                  </xsd:sequence>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:complexType name="BatchData">
            <xsd:sequence>
               <xsd:choice>
                  <xsd:element name="textDataFile" type="xsd:base64Binary"
xmime:expectedContentTypes="text/sms"/>
               </xsd:choice>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="Recipients">
            <xsd:sequence>
               <xsd:element maxOccurs="unbounded" name="recipient" type="xsd:string"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="MessageIds">
            <xsd:sequence>
               <xsd:element maxOccurs="unbounded" name="messageId" type="xsd:string"/>
```

```
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="BatchMessageStatus">
            <xsd:sequence>
               <xsd:element name="statusCode" type="xsd:int"/>
               <xsd:element name="statusText" type="xsd:string"/>
               <xsd:element minOccurs="0" name="batchId" type="xsd:string"/>
               <xsd:element minOccurs="0" name="batchReferenceId" type="xsd:string"/>
               <xsd:element minOccurs="0" name="batchMessageReferenceId" type=
"xsd:string"/>
               <xsd:element minOccurs="0" name="id" type="xsd:string"/>
               <xsd:element minOccurs="0" name="sender" type="xsd:string"/>
               <xsd:element name="recipient" type="xsd:string"/>
               <xsd:element minOccurs="0" name="conversationId" type="xsd:string"/>
               <xsd:element name="time" type="xsd:dateTime"/>
               <xsd:element minOccurs="0" name="attributes" type="Attributes"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="Attributes">
            <xsd:sequence>
               <xsd:element maxOccurs="unbounded" name="attribute" type="Attribute"/>
            </xsd:sequence>
         </xsd:complexType>
         <xsd:complexType name="Attribute">
            <xsd:sequence>
               <xsd:element name="name" type="xsd:string"/>
               <xsd:element 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:element name="date" type="xsd:dateTime"/>
            </xsd:choice>
         </xsd:complexType>
         <xsd:complexType name="ErrorDetail">
            <xsd:sequence>
               <xsd:element name="errorCode" type="xsd:int"/>
               <xsd:element minOccurs="0" name="reasonCode" type="xsd:int"/>
               <xsd:element minOccurs="0" name="errorDescription" type="xsd:string"/>
            </xsd:sequence>
         </xsd:complexType>
      </xsd:schema>
   </wsdl:types>
   <wsdl:message name="BatchMessageIdRequest">
```

```
      <wsdl:part element="tns:BatchMessageIdRequest" name="BatchMessageIdRequest"/>
   </wsdl:message>
   <wsdl:message name="BatchMessageIdFault">
      <wsdl:part element="tns:BatchMessageIdFault" name="BatchMessageIdFault"/>
   </wsdl:message>
   <wsdl:message name="BatchInfoRequest">
      <wsdl:part element="tns:BatchInfoRequest" name="BatchInfoRequest"/>
   </wsdl:message>
   <wsdl:message name="BatchMessageStatusFault">
      <wsdl:part element="tns:BatchMessageStatusFault" name="BatchMessageStatusFault
"/>
   </wsdl:message>
   <wsdl:message name="SendBatchFault">
      <wsdl:part element="tns:SendBatchFault" name="SendBatchFault"/>
   </wsdl:message>
   <wsdl:message name="SendBatchRequest">
      <wsdl:part element="tns:SendBatchRequest" name="SendBatchRequest"/>
   </wsdl:message>
   <wsdl:message name="SendBatchResponse">
      <wsdl:part element="tns:SendBatchResponse" name="SendBatchResponse"/>
   </wsdl:message>
   <wsdl:message name="BatchMessageStatusResponse">
      <wsdl:part element="tns:BatchMessageStatusResponse" name=
"BatchMessageStatusResponse"/>
   </wsdl:message>
   <wsdl:message name="BatchInfoResponse">
      <wsdl:part element="tns:BatchInfoResponse" name="BatchInfoResponse"/>
   </wsdl:message>
   <wsdl:message name="BatchMessageStatusRequest">
      <wsdl:part element="tns:BatchMessageStatusRequest" name=
"BatchMessageStatusRequest"/>
   </wsdl:message>
   <wsdl:message name="BatchInfoFault">
      <wsdl:part element="tns:BatchInfoFault" name="BatchInfoFault"/>
   </wsdl:message>
   <wsdl:message name="BatchMessageIdResponse">
      <wsdl:part element="tns:BatchMessageIdResponse" name="BatchMessageIdResponse"/>
   </wsdl:message>
   <wsdl:portType name="messaging-batch-v1">
      <wsdl:operation name="BatchMessageId">
         <wsdl:input message="tns:BatchMessageIdRequest" name="BatchMessageIdRequest
"/>
         <wsdl:output message="tns:BatchMessageIdResponse" name=
"BatchMessageIdResponse"/>
         <wsdl:fault message="tns:BatchMessageIdFault" name="BatchMessageIdFault"/>
      </wsdl:operation>
      <wsdl:operation name="BatchInfo">
         <wsdl:input message="tns:BatchInfoRequest" name="BatchInfoRequest"/>
         <wsdl:output message="tns:BatchInfoResponse" name="BatchInfoResponse"/>
         <wsdl:fault message="tns:BatchInfoFault" name="BatchInfoFault"/>
      </wsdl:operation>
```

```
      <wsdl:operation name="BatchMessageStatus">
         <wsdl:input message="tns:BatchMessageStatusRequest" name=
"BatchMessageStatusRequest"/>
         <wsdl:output message="tns:BatchMessageStatusResponse" name=
"BatchMessageStatusResponse"/>
         <wsdl:fault message="tns:BatchMessageStatusFault" name=
"BatchMessageStatusFault"/>
      </wsdl:operation>
      <wsdl:operation name="SendBatch">
         <wsdl:input message="tns:SendBatchRequest" name="SendBatchRequest"/>
         <wsdl:output message="tns:SendBatchResponse" name="SendBatchResponse"/>
         <wsdl:fault message="tns:SendBatchFault" name="SendBatchFault"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="messaging-batch-v1Soap12" type="tns:messaging-batch-v1">
      <soap12:binding style="document" transport="
http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="BatchMessageId">
         <soap12:operation soapAction=""/>
         <wsdl:input name="BatchMessageIdRequest">
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchMessageIdResponse">
            <soap12:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchMessageIdFault">
            <soap12:fault name="BatchMessageIdFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="BatchInfo">
         <soap12:operation soapAction=""/>
         <wsdl:input name="BatchInfoRequest">
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchInfoResponse">
            <soap12:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchInfoFault">
            <soap12:fault name="BatchInfoFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="BatchMessageStatus">
         <soap12:operation soapAction=""/>
         <wsdl:input name="BatchMessageStatusRequest">
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchMessageStatusResponse">
            <soap12:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchMessageStatusFault">
            <soap12:fault name="BatchMessageStatusFault" use="literal"/>
```

```
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="SendBatch">
         <soap12:operation soapAction=""/>
         <wsdl:input name="SendBatchRequest">
            <soap12:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="SendBatchResponse">
            <soap12:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="SendBatchFault">
            <soap12:fault name="SendBatchFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:binding name="messaging-batch-v1Soap11" type="tns:messaging-batch-v1">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http
"/>
      <wsdl:operation name="BatchMessageId">
         <soap:operation soapAction=""/>
         <wsdl:input name="BatchMessageIdRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchMessageIdResponse">
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchMessageIdFault">
            <soap:fault name="BatchMessageIdFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="BatchInfo">
         <soap:operation soapAction=""/>
         <wsdl:input name="BatchInfoRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchInfoResponse">
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchInfoFault">
            <soap:fault name="BatchInfoFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="BatchMessageStatus">
         <soap:operation soapAction=""/>
         <wsdl:input name="BatchMessageStatusRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="BatchMessageStatusResponse">
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="BatchMessageStatusFault">
```

```
            <soap:fault name="BatchMessageStatusFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="SendBatch">
         <soap:operation soapAction=""/>
         <wsdl:input name="SendBatchRequest">
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="SendBatchResponse">
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="SendBatchFault">
            <soap:fault name="SendBatchFault" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="messaging-batch-v1Service">
      <wsdl:port binding="tns:messaging-batch-v1Soap12" name="messaging-batch-
v1Soap12">
         <soap12:address location="http://localhost:8080/batch/messaging-batch-v1"/>
      </wsdl:port>
      <wsdl:port binding="tns:messaging-batch-v1Soap11" name="messaging-batch-
v1Soap11">
         <soap:address location="http://localhost:8080/batch/messaging-batch-v1"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```