

## **1. Introduction**

### **SCOPE**

This document specifies VOIP services system interface requests and responses sent between partner application and Telia's VOIP services platform using HTTP/SOAP/XML as bearer for messages (HTTP POST message).

## **2. General Information**

### **INTRODUCTION**

The interface allows third party systems to carry out basic tasks like click to call, fetch group phone numbers, fetch call history, modify phonebook etc.

The interface conforms to SOAP standard as specified in the W3C recommendations.

The interface is simple SOAP RPC-style request and response over http(http1.1) using POST message.

### **BASE PROTOCOL USAGE**

#### **2.1.1 TRANSPORT**

TCP transport protocol shall be used for http.

#### **2.1.2 SECURITY**

TLS https must be used for http.

OASIS web server security is used for authentication(soap/xml level).

## **3. Procedures**

Partner application makes requests towards services management interface using SOAP.

Services management interface responds with SOAP response message.

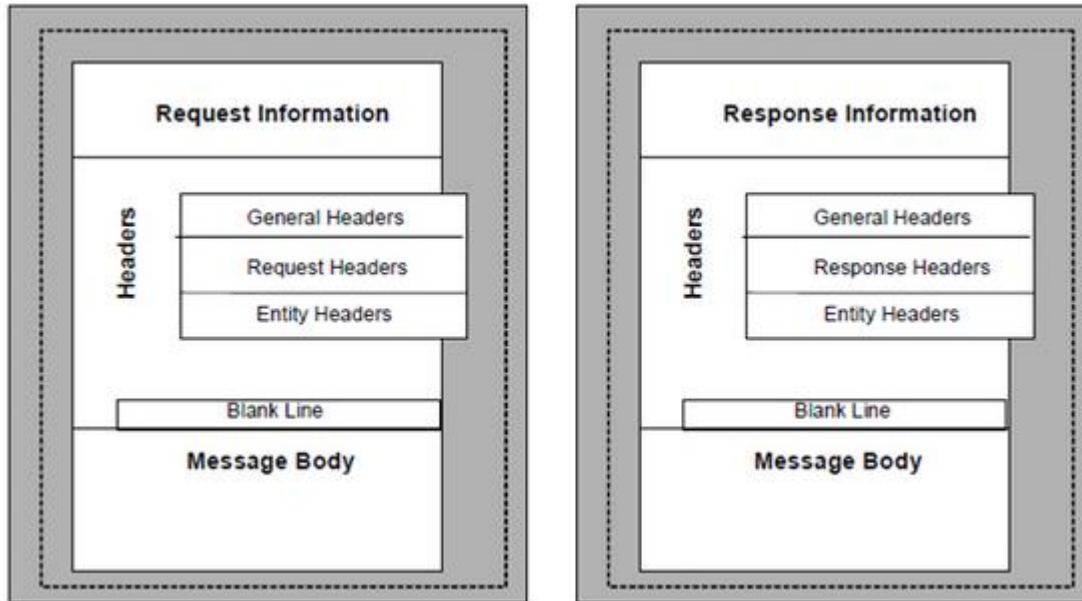
The SOAP messages are carried within http POST message and http response code 200 is used to carry the successful response on the message.

## **4. Message format**

### **HTTP**

#### **4.1.1 GENERAL**

The HTTP client (provisioning system) is responsible for initiating the communication with the server. To do this it sends a formatted request and waits for a response from the server.



The request information includes the method being used to make the request (in this case the POST message), the URI of the requested resource, and the version of the HTTP (HTTP/1.1):

General header information is used in both requests and responses and includes information such as the date, connection, or caching information.

Request header information is used to specify relevant information about the client for the server such as the data it prefers to receive, any conditions to the request, or the maximum number of timers the request can be forwarded.

The Entity header information is used to specify information about the body (entity) of the request or response sent, such as the type of data, the length etc.

The Blank Line always follows the request header to indicate the end of the header information, and possibly the start of the body of the request.

The Body of the request may contain POST method parameters, files PUT on the server, other files, java data etc. In our case the body contains the SOAP/XML information.

The Response information includes the HTTP version of the response, the status code indicating the result of the request and the message associated with the status code (for example a successful request: HTTP/1.1 200 OK).

Response headers are specific to the server response and are used to include server-specific information such as cookie setting, or authentication.

#### **4.1.2 OASIS WEB SERVER SECURITY AUTHENTICATION**

Oasis web server authentication information is in SOAP header

Username and password must be group administrator's username and password.

While using soapUI for development WSS-Password Type must be PasswordText

### 4.1.3 SOAP 1.1 BINDING

When using HTTP for SOAP1.1, one extra HTTP header is added: the 'SOAPAction' header. The 'SOAPAction' header provides the intent of the message - it allows servers and firewalls to determine how to handle, filter, or route the SOAP request appropriately. This header will have a value of an empty pair of quotes. The empty quotes means that the URI information is carried in the HTTP request URI. The body of the message contains the XML document describing the SOAP request.

### 4.1.4 HTTP ERRORS

HTTP status codes are used by the HTTP server in the response to indicate the result of the request. If status code is not 200, partner application must assume that operation was unsuccessful.

Example: HTTP/1.1 404 Not Found

## SOAP/XML

### 4.1.5 GENERAL

The simple Object Access Protocol (SOAP) is an XML-based protocol that allows the exchange of information in a decentralized, distributed environment. A SOAP message is an ordinary XML document containing the following elements:

- A required Envelope element that identifies the XML document as a SOAP message
- An optional Header element that contains header information
- A required Body element that contains call and response information
- An optional Fault element that provides information about errors that occurred while processing the message

All the elements are declared in the default namespace for the SOAP envelope <http://schemas.xmlsoap.org/soap/envelope/> (SOAP 1.1) and the default namespace for SOAP encoding and data types is: <http://schemas.xmlsoap.org/soap/encoding/> (SOAP 1.1). The operations and methods to use towards provisioning interface are defined in different wsdl-s.

### 4.1.6 WSDL SPECIFICATION

#### 4.1.6.1 SERVICES MANAGEMENT

All the available operations are listed in <https://teenused.telia.ee/external/services/index.php>

WSDL is available here <https://teenused.telia.ee/external/services/index.php?wsdl>

##### 4.1.6.1.1 clickToCall

An operation for initiating voice Calls.

Inputs:

- Anumber – Calling party's phone number: must be in E.164 format, with country code, but without prefix + e.g. 3726200001
- Bnumber – Called party's phone number – doesn't have to be in E.164 format

#### 4.1.6.1.2 activateNotification

An operation for activating notifications of VOIP events to given phone number. VOIP events are: incoming call and missed call. Event notifications are sent to given URL via HTTP GET.

Example of GET request: [http://exampel.com/notify.php?](http://exampel.com/notify.php?ts=1328150351&anr=37262000001&bnr=372620000002&fn=Turvateenistus+Alfastar+AS&type=Start&answer=No)

[ts=1328150351&anr=37262000001&bnr=372620000002&fn=Turvateenistus+Alfastar+AS&type=Start&answer=No.](http://exampel.com/notify.php?ts=1328150351&anr=37262000001&bnr=372620000002&fn=Turvateenistus+Alfastar+AS&type=Start&answer=No)

GET parameters:

- ts – unix timestamp
- anr - Calling party's phone number
- bnr - Called party's phone number
- fn - Calling party's name
- type – event type
- answer – answer indicator (yes/no)

Inputs:

- Anumber – phone number to activate: must be in E.164 format, with country code, but without prefix + e.g. 3726200001
- TargetURL – An URL where to send notification requests via HTTP GET. Example: (<http://exampel.com/notify.php>)

IP address of the notification server for outgoing traffic is 62.65.42.130. All notification events are sent from that IP address.

When notifications are not reaching your IS, when they should, check:

Notification URL. Verify that notification URL is correct

Web server logs. Check that requests from IP 62.65.42.130 are reaching your web server and that the request is in correct format.

Firewall settings. Check that Firewall allows incoming traffic from IP 62.65.42.130 to your web server.

#### 4.1.6.1.3 listOwnNumbers

Retrieves all phone numbers in given group

Inputs: -

#### 4.1.6.1.4 updateCallingLineNameIdentity

Updates Display Name information associated with a phone number

Inputs:

- Anumber - phone number: must be in E.164 format, with country code, but without prefix + e.g. 3726200001
- FirstName – first name
- LastName – last name

#### 4.1.6.1.5 getCallHistory

Retrieves Call logs for last 31 days for given phone number.

Inputs:

- Anumber – phone number: must be in E.164 format, with country code, but without prefix + e.g. 3726200001

#### 4.1.6.1.6 deactivateNotification

Deactivate notification messages for given phone number. This operation removes association between target URL, associated by activateNotification operation, and phone number

Inputs:

- Anumber – phone number: must be in E.164 format, with country code, but without prefix + e.g. 3726200001

### **4.1.6.2 PHONEBOOK MANAGEMENT**

All the available operations are listed in

<https://teenused.telia.ee/external/phonebook/index.php> WSDL is available here

<https://teenused.telia.ee/external/phonebook/index.php?wsdl>

Phonebook can be used as a normal companywide phonebook for company members. The phonebook will also be used to insert name info into incoming calls' information as Calling Party Name Identity.

#### 4.1.6.2.1 AddEntry

Add an entry to your company wide phonebook

Inputs:

- DisplayName – Name displayed as Calling Party Name Identity
- PhoneNumber1 – phone number associated with DisplayName
- PhoneNumber2 - phone number associated with DisplayName

#### 4.1.6.2.2 updateEntry

Inputs:

- DisplayName – Name displayed as Calling Party Name Identity
- PhoneNumber1 – phone number associated with DisplayName
- PhoneNumber2 - phone number associated with DisplayName
- Id – ID of the entry you wish to modify

#### 4.1.6.2.3 deleteEntry

Remove an entry in your company wide phonebook

Inputs:

- id – ID of the entry you wish to remove

#### 4.1.6.2.4 listPhoneBookEntries

Retrieves all phonebook entries.

Inputs: -

## **5. Information elements**

### **GENERAL SOAP ELEMENTS**

#### **5.1.1 ENVELOPE**

The mandatory Envelope header provides the root element for the XML document, and is used to include namespace declarations and the Header and Body elements.

#### **5.1.2 SOAP ENVELOPE NAMESPACE**

The envelope element resides in the SOAP envelope namespace. Different namespaces are used for SOAP1.1 and SOAP1.2.

Example SOAP1.1

<http://schemas.xmlsoap.org/soap/envelope/>

#### **5.1.3 HEADER**

A SOAP header element is optional. A SOAP header is an extension mechanism that provides a way to pass information in SOAP messages that is not application payload. Such "control" information includes, for example, passing directives or contextual information related to the processing of the message.

The contents of the header element are application defined and not a part of the SOAP specifications

Example: env:Header

#### **5.1.4 BODY**

The SOAP body is the mandatory element within the SOAP env:Envelope, which implies that this is where the main end-to-end information conveyed in a SOAP message must be carried.

The contents of the body element are application defined and not a part of the SOAP specifications

The actual request/response messages are made up using direct children of the Body element (which are called body entries).

Example: env:Body

## PARAMETERS IN SOAP HEADER AND BODY ELEMENTS

### 5.1.5 General

The latest version of parameters and types in use for provisioning interface are available in the wsdl's(see paragraph 4.2.2).

## SOAP ERROR RESPONSES

### 5.1.6 GENERAL

If TAG is present in response, it means operation was unsuccessful. ServiceError consists of machine readable ErrorCode and human readable ErrorMessage.

Soap-level errors do not trigger http-level error responses. This means soap-errors are responded within 200 OK http-level responses.

#### [Example 1: ServiceError](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <getUserResponse xmlns="http://teenused.telia.ee/external/soap/user">
      <ServiceError>
        <ErrorCode>ERR001</ErrorCode>
        <ErrorMessage>Aiauthentication error</ErrorMessage>
        <ErrorDescription/>
      </ServiceError>
    </getUserResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### [Example 2: ServiceError](#)

```
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ServiceError>
      <ErrorCode xsi:type="xsd:string">SOAP-ENV:Client</ErrorCode>
      <ErrorMessage xsi:type="xsd:string">error in msg parsing: XML error
parsing SOAP payload on line 7: Mismatched tag</ErrorMessage>
      <ErrorDescription xsi:type="xsd:string">error in msg parsing: XML error
parsing SOAP payload on line 7: Mismatched tag</ErrorDescription>
    </ServiceError>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

If service error is empty or ErrorCode and ErrorMessage are empty you must assume that operation was successful.

For successful information requests (getCallHistory, listOwnNumbers etc) additional response tags are available as described in WSDL by ResponseType

### Example: response type for listOwnNumbers

```
<xsd:complexType name="listOwnNumbersResponseType">
  <xsd:all>
  <xsd:element name="ServiceError" type="tns:ServiceError"/>
  <xsd:element name="listofNumbers" type="tns:listofNumbers"/>
  </xsd:all>
</xsd:complexType&
```

### Example of SOAP response for listOwnNumbers

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <listOwnNumbersResponse
xmlns="http://teenused.telia.ee/external/services/">
      <listofNumbers>
        <item>
          <Number>6403231</Number>
        </item>
        <item>
          <Number>6403232</Number>
        </item>
      </listofNumbers>
    </listOwnNumbersResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## 5.1.7 SOAP ERROR DETAILS

So far the main reasons of errors have been:

Authentication has done using random usernames instead of main user's. Main user's username example is: Prandomnumber@elion.ee (e.g. P0000000@elion.ee).

The phone number used has been in a wrong format. It should be as follows: 372xxxxxxx

Error code	ErrorMessage	ErrorDescription
ERR001	Authentication failed	Account failure
ERR002	Missing Element: Element Name	Mandatory element missing
ERR003	Authorization Error	Requested operation is not permitted
ERR1001	Dialing Error	Call could not be made
ERR1002	Unknown ID	In operation a nonexistent element ID was used
ERR1003	Unallowed Character	Request contained an unallowed Character. Probably in the name field. Allowed Characters are : ASCII characters plus õ,ü,ö,ä.

## 6. Usage Examples

### CLICK TO CALL

#### [Request](#)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://teenused.telia.ee/external/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:clickToCall>
      <ser:clickToCall>
        <!--You may enter the following 2 items in any order-->
        <ser:Anumber>3726401234</ser:Anumber>
        <ser:Bnumber>6401235</ser:Bnumber>
      </ser:clickToCall>
    </ser:clickToCall>
  </soapenv:Body>
</soapenv:Envelope>
```

#### [OK response](#)

```
<xsd:complexType name="listOwnNumbersResponseType">
  <xsd:all>
    <xsd:element name="ServiceError" type="tns:ServiceError"/>
    <xsd:element name="listofNumbers" type="tns:listofNumbers"/>
  </xsd:all>
</xsd:complexType&
```

#### [Error response](#)

```
<xsd:complexType name="listOwnNumbersResponseType">
  <xsd:all>
    <xsd:element name="ServiceError" type="tns:ServiceError"/>
    <xsd:element name="listofNumbers" type="tns:listofNumbers"/>
  </xsd:all>
</xsd:complexType&
```

### ACTIVATE NOTIFICATION

#### [Request](#)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://teenused.telia.ee/external/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:activateNotification>
      <ser:activate>
        <!--You may enter the following 2 items in any order-->
        <ser:Anumber>37264012345</ser:Anumber>
        <ser:TargetURL>http://excmpl.com/notify.php</ser:TargetURL>
      </ser:activate>
    </ser:activateNotification>
  </soapenv:Body>
</soapenv:Envelope>
```

#### [Response](#)

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <activateNotificationResponse
xmlns="http://teenused.telia.ee/external/services/">
      <ServiceError>
        <ErrorCode/>
        <ErrorMessage/>
        <ErrorDescription/>
      </ServiceError>
    </activateNotificationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## DEACTIVATE NOTIFICATION

### [Request](#)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://teenused.telia.ee/external/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:deactivateNotification>
      <ser:Anumber>
        <ser:Anumber>37264012345</ser:Anumber>
      </ser:Anumber>
    </ser:deactivateNotification>
  </soapenv:Body>
</soapenv:Envelope>

```

### [Response](#)

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <deactivateNotificationResponse
xmlns="http://teenused.telia.ee/external/services/">
      <ServiceError>
        <ErrorCode/>
        <ErrorMessage/>
        <ErrorDescription/>
      </ServiceError>
    </deactivateNotificationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## GET LIST OF OWN NUMBERS

### [Request](#)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://teenused.telia.ee/external/services/">
  <soapenv:Header/>
  <soapenv:Body>

```

```
<ser:listOwnNumbers/>
</soapenv:Body>
</soapenv:Envelope>
```

### [Response](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <listOwnNumbersResponse
xmlns="http://teenused.telia.ee/external/services/">
      <listofNumbers>
        <item>
          <Number>6403231</Number>
        </item>
        <item>
          <Number>6403232</Number>
        </item>
      </listofNumbers>
    </listOwnNumbersResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## GET CALL HISTORY

### [Request](#)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://teenused.telia.ee/external/services/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:getCallHistory>
      <ser:Anumber>
        <ser:Anumber>64012347</ser:Anumber>
      </ser:Anumber>
    </ser:getCallHistory>
  </soapenv:Body>
</soapenv:Envelope>
```

### [Response](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <getCallHistoryResponse
xmlns="http://teenused.telia.ee/external/services/">
      <listofCallDetailRecord>
        <item>
          <CallDetailRecord>
            <id>50762615</id>
            <CallingNumber>6000001</CallingNumber>
            <CalledNumber>6401234</CalledNumber>
            <AnswerIndic>Yes</AnswerIndic>
            <Direction>Terminating</Direction>
            <StartTime>2011-11-29T12:19:18</StartTime>
            <OtherPartyName># #</OtherPartyName>
          </CallDetailRecord>
        </item>
      </listofCallDetailRecord>
    </getCallHistoryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    <AnswerTime>2011-11-29T12:19:21</AnswerTime>
    <ReleaseTime>2011-11-29T12:19:36</ReleaseTime>
  </CallDetaileRecord>
</item>
<item>
  <CallDetaileRecord>
    <id>24297107</id>
    <CallingNumber>60000002</CallingNumber>
    <CalledNumber>60000003</CalledNumber>
    <AnswerIndic>No</AnswerIndic>
    <Direction>Terminating</Direction>
    <StartTime>2011-10-31T09:30:07</StartTime>
    <OtherPartyName>A Client</OtherPartyName>
  </CallDetaileRecord>
</item>
</listofCallDetaileRecord>
</getCallHistoryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## UPDATE CALLING LINE NAME IDENTITY

### [Response](#)

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <updateCallingLineNameIdentityResponse
xmlns="http://teenused.telia.ee/external/services/">
      <ServiceError>
        <ErrorCode>ERR002</ErrorCode>
        <ErrorMessage>Missin Element: LastName</ErrorMessage>
        <ErrorDescription/>
      </ServiceError>
    </updateCallingLineNameIdentityResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## PHONEBOOK: ADD ENTRY

### [Request](#)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:phon="http://teenused.telia.ee/external/phonebook/">
  <soapenv:Header/>
  <soapenv:Body>
    <phon:AddEntry>
      <phon:PhoneBookEntry>
        <!--You may enter the following 4 items in any order-->
        <phon:DisplayName>Jhon</phon:DisplayName>
        <phon:PhoneNumber1>55512345</phon:PhoneNumber1>
        <phon:PhoneNumber2>6012345</phon:PhoneNumber2>
        <phon:id>?</phon:id>
      </phon:PhoneBookEntry>
    </phon:AddEntry>
  </soapenv:Body>

```

```
</soapenv:Envelope>
```

#### [Response](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-  
ENC="http://schemas.xmlsoap.org/soap/encoding/">  
  <SOAP-ENV:Body>  
    <AddEntryResponse xmlns="http://teenused.telia.ee/external/phonebook/">  
      <ServiceError/>  
    </AddEntryResponse>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

## PHONEBOOK: DELETE ENTRY

#### [Request](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-  
ENC="http://schemas.xmlsoap.org/soap/encoding/">  
  <SOAP-ENV:Body>  
    <phon:deleteEntry>  
      <phon:PhoneBookEntry>  
        <phon:id>12</phon:id>  
      </phon:PhoneBookEntry>  
    </phon:deleteEntry>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

#### [Response](#)

```
<SOAP-ENV:Envelope xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-  
ENC="http://schemas.xmlsoap.org/soap/encoding/">  
  <SOAP-ENV:Body>  
    <phon:deleteEntry>  
      <phon:PhoneBookEntry>  
        <phon:id>12</phon:id>  
      </phon:PhoneBookEntry>  
    </phon:deleteEntry>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

## PHONEBOOK: UPDATE ENTRY

#### [Request](#)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:phon="http://teenused.telia.ee/external/phonebook/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <phon:updateEntry>  
      <phon:PhoneBookEntry>  
        <!--You may enter the following 4 items in any order-->
```

```

    <phon:DisplayName>Jhon</phon:DisplayName>
    <phon:PhoneNumber1>55112233</phon:PhoneNumber1>
    <phon:PhoneNumber2>6123456</phon:PhoneNumber2>
    <phon:id>1</phon:id>
  </phon:PhoneBookEntry>
</phon:updateEntry>
</soapenv:Body>
</soapenv:Envelope>

```

#### [Response](#)

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <updateEntryResponse
xmlns="http://teenused.telia.ee/external/phonebook/">
      <ServiceError/>
    </updateEntryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## PHONEBOOK: LIST ENTRIES

#### [Request](#)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:phon="http://teenused.telia.ee/external/phonebook/">
  <soapenv:Header/>
  <soapenv:Body>
    <phon:listPhoneBookEntries/>
  </soapenv:Body>
</soapenv:Envelope>

```

#### [Response](#)

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:phon="http://teenused.telia.ee/external/phonebook/">
  <soapenv:Header/>
  <soapenv:Body>
    <phon:listPhoneBookEntries/>
  </soapenv:Body>
</soapenv:Envelope>

```

## DESCRIPTION OF EVENT-MESSAGE

This is a brief description on how to realize a popup in a browser with Telia's telephony system and notification messages.

## COMPONENTS

Web/core server - normal web server with most of the business logic e.g. CRM application server

COMET server – a server that realizes AJAX push technology: also known as COMET technology

API server – a RPC server that receives automated requests from external or internal components  
Telephony – TELIA's telephony system  
User – User's browser

## SEQUENCE OF EVENTS

Request a web page – normal http/https request to application server  
Response – response to the previous request: includes normal http response plus an order to open a connection to COMET server  
http request to comet server – normal http/https request to COMET server. Except response may come in minutes or hours.  
http get - http get from Telia' to inform partner's API server of incoming event  
Check session – API server checks from application if a user with phone number in event has an active session  
Response - Application responds with session and some data  
API server informs the comet server to push some data to the user with given session  
Response – response to request in p3 includes some data and JS code  
A popup is displayed on user's browser

## 7. References

HTTP Protocol RFC 2616

SOAP1.1 Specification <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>

WSDL Specification, <http://www.w3.org/TR/2007/WD-wsdl20-20070326>

XML Specification, <http://www.w3.org/TR/REC-xml/>  
<http://schemas.xmlsoap.org/soap/envelope/>