



**swisscom**

From Swisscom (Schweiz) AG

Issue date 12<sup>nd</sup> May 2008

Subject Broadband Connectivity Service and Full Access (T-)TAL  
Appointment Booking

## B2B Appointment Booking Specification

Status	released
Contract Version	-
WSG Version	10
Valid until	recalled
Server location	\\SMEF02\WSG\0_Member\WSG0\Dokumente

**Swisscom (Schweiz) AG**  
CH-3050 Bern

Title  
Section  
Valid from  
Issue date

B2B Appointment Booking Specification  
**Fehler! Verweisquelle konnte nicht gefunden werden.**  
**Fehler! Verweisquelle konnte nicht gefunden werden.**  
12nd May 2008

Version **Fehler! Verweisquelle konnte nicht gefunden werden.**  
Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**  
**Fehler! Verweisquelle konnte nicht gefunden werden.**

### Checklist of changes

Version	Date	Changed by	Comments / nature of the change
01	02.05.2008	HP	First Edition

### Release

int.Version	Date	Released by	Comments / nature of the change
10	23.06.2008	SCS Rolf Seltmann	Document released for internal use only

## Contents

<b>Checklist of changes .....</b>	<b>2</b>
<b>Contents .....</b>	<b>2</b>
<b>1 Introduction .....</b>	<b>4</b>
1.1 Purpose .....	4
1.2 Scope .....	4
1.3 Target readership, requirements of the reader .....	4
1.4 Terms and abbreviations .....	4
1.5 Referenced documents .....	4
1.6 Single Point of Contact in Swisscom (SPOC) .....	4
1.7 Document updates .....	4
<b>2 Business Processes .....</b>	<b>6</b>
<b>3 WSG Model - Appointment Booking .....</b>	<b>7</b>
3.1 Operations.....	7
3.2 Messages.....	7
3.3 Types.....	8
<b>4 Web Service Interface.....</b>	<b>13</b>
4.1 Security .....	13
4.1.1 Encryption .....	13
4.1.2 Authentication, Authorization, and Accounting (AAA).....	13
4.2 Model: WSDL and XML Schemas .....	13
4.3 The WSG Appointment Booking Outbound Web Service .....	13
4.3.1 Success or Error status .....	14
4.3.2 Error Codes .....	14
General Web Service Exceptions that may occur in all functions.....	14
Specific Exceptions for the function <i>getAvailableTimeSlots</i> .....	14
Specific Exceptions for the function <i>bookAppointment</i> .....	14
Specific Exceptions for the function <i>rescheduleAppointment</i> .....	14
Specific Exceptions for the function <i>cancelAppointment</i> .....	14
4.4 Connection Testing.....	15
<b>5 Appendix A .....</b>	<b>16</b>
5.1 Securing a WEB-Service with Powergate.....	16
5.2 Setting up a new WEB-Service .....	16
5.3 Sample Client with wss4j .....	16
5.4 Errors.....	19
5.5 Login from the user point of view .....	20

## 1 Introduction

The WSG Appointment Booking business processes can be used via a GUI interface or via web service interface.

### 1.1 Purpose

This document provides the detailed technical specification for the implementation of these interfaces.

### 1.2 Scope

In summary, for the Appointment Booking business processes, the specification provides:

- The detail specification of the web service

### 1.3 Target readership, requirements of the reader

This document addresses the user and the developer of a client application using the mentioned services.

### 1.4 Terms and abbreviations

BB	BroadBand
TT	Trouble Ticket
CUG	Closed User Group
GUI	Graphical User Interface
PTS	Provider of Telecom Services
SSH	Secure Shell
WSG	Web Service Gateway

### 1.5 Referenced documents

None.

### 1.6 Single Point of Contact in Swisscom (SPOC)

The SPOC for all questions or suggestions related to the present document is:  
ServiceDesk.IT@swisscom.com

In the near future Swisscom will create a closed user group (CUG) for the PTS on the internet which will contain information and FAQs about the described services.

### 1.7 Document updates

In case of changes Swisscom will provide the PTS with the latest update of this document at least three weeks before the changes becomes effective.



As soon as the CUG mentioned above is available, updates of this document will be communicated via the CUG.

## 2 Business Processes

This business processes provided by this interface are described by the following XML-schema:



wsgAppBooking\_v1.zip

This ZIP-File contains:

- Readme.txt : general information on this file (contents)
- ChangeLog.txt : the details of the changes per released version (change-history).
- schema/\* : the schema definition as well as the web-service definition
- doc/
  - wsgAb.xsd.html : an HTML-based documentation of the schema.
  - wsgAbOutbound.wsdl.html : an HTML-based documentation of the outbound web-service.
- samples/
  - file/\* : sample XML-instances for all upload- and download- files.

(Remark: refer to file attachment if this document is in PDF format)

The following table provides an overview of the available business processes for Appointment Booking:

Business Process	Ver.	WebService / Operation <i>WsgAppBookVxxx</i>
GET_TIMESLOTS	V001	getAvailableTimeSlots
BOOK_APPOINTMENT	V001	bookAppointment
RESCHEDULE_APPOINTMENT	V001	rescheduleAppointment
CANCEL_APPOINTMENT	V001	cancelAppointment

### 3 WSG Model - Appointment Booking

#### 3.1 Operations

##### 3.1.1.1 getAvailableTimeSlots

**Description:** This operation returns the available time slots for an appointment

Input message	Output message
<a href="#">getAvailableTimeSlotsRequest</a>	<a href="#">getAvailableTimeSlotsResponse</a>

##### 3.1.1.2 bookAppointment

**Description:** This operation creates an appointment.

Input message	Output message
<a href="#">bookAppointmentRequest</a>	<a href="#">bookAppointmentResponse</a>

##### 3.1.1.3 rescheduleAppointment

**Description:** This operation modifies an existing appointment (reschedule)..

Input message	Output message
<a href="#">rescheduleAppointmentRequest</a>	<a href="#">rescheduleAppointmentResponse</a>

##### 3.1.1.4 cancelAppointment

**Description:** This operation cancels an existing appointment..

Input message	Output message
<a href="#">cancelAppointmentRequest</a>	<a href="#">cancelAppointmentResponse</a>

#### 3.2 Messages

##### 3.2.1.1 getAvailableTimeSlotsRequest

Element	Type	Occ	Comment
parameters	<a href="#">getAvailableTimeSlots</a>	1..1	

##### 3.2.1.2 getAvailableTimeSlotsResponse

Element	Type	Occ	Comment
parameters	<a href="#">getAvailableTimeSlotsResponse</a>	1..1	

##### 3.2.1.3 bookAppointmentRequest

Element	Type	Occ	Comment
parameters	<a href="#">bookAppointment</a>	1..1	

##### 3.2.1.4 bookAppointmentResponse

Element	Type	Occ	Comment
parameters	<a href="#">bookAppointmentResponse</a>	1..1	

### 3.2.1.5 rescheduleAppointmentRequest

Element	Type	Occ	Comment
parameters	<a href="#">rescheduleAppointment</a>	1..1	

### 3.2.1.6 rescheduleAppointmentResponse

Element	Type	Occ	Comment
parameters	<a href="#">bookAppointmentResponse</a>	1..1	

### 3.2.1.7 cancelAppointmentRequest

Element	Type	Occ	Comment
parameters	<a href="#">cancelAppointment</a>	1..1	

### 3.2.1.8 cancelAppointmentResponse

Element	Type	Occ	Comment
parameters	<a href="#">cancelAppointmentResponse</a>	1..1	

## 3.3 Types

### 3.3.1.1 getAvailableTimeSlots

Element	Type	Occ	Comment
request	<a href="#">getAvailableTimeSlotsRequestType</a>	1..1	Purpose: Get the available timeslots for an appointment within the range of fromDate and toDate.

### 3.3.1.2 getAvailableTimeSlotsResponse

Element	Type	Occ	Comment
response	<a href="#">getAvailableTimeSlotsResponseType</a>	1..1	Purpose: Get the available timeslots for an appointment.

### 3.3.1.3 bookAppointment

Element	Type	Occ	Comment
request	<a href="#">bookAppointmentRequestType</a>	1..1	Purpose: Book a time slot for an appointment.

### 3.3.1.4 bookAppointmentResponse

Element	Type	Occ	Comment
response	<a href="#">bookAppointmentResponseType</a>	1..1	Purpose: Response to appointment booking or moving (reschedule) request.

### 3.3.1.5 rescheduleAppointment

Element	Type	Occ	Comment
request	<a href="#">rescheduleAppointmentRequestType</a>	1..1	Purpose: Update the time and date of an appointment (reschedule). Books a new appointment and cancels the old one at the same time.



### 3.3.1.6 cancelAppointment

Element	Type	Occ	Comment
request	<a href="#">cancelAppointmentRequestType</a>	1..1	Purpose: Cancel an appointment.

### 3.3.1.7 cancelAppointmentResponse

Element	Type	Occ	Comment
response	<a href="#">cancelAppointmentResponseType</a>	1..1	Purpose: Response to appointment cancel request.

### 3.3.1.8 getAvailableTimeSlotsRequestType

**Description:** Purpose: Get the available timeslots for an appointment within the range of fromDateTime and toDateTime.

**Used by:** [getAvailableTimeSlots](#)

**Extension of:** [requestType](#)

Element	Type	Occ	Comment												
from-DateTime	xs:dateTime	0..1	Beginning of requested appointment period. Defaults to current date and time if not specified.												
toDateTime	xs:dateTime	0..1	End of requested appointment period. Deraults to current date and time + 10 days if not specified												
Choice	<table border="1"> <thead> <tr> <th>Element</th> <th>Type</th> <th>Occ</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>troubleTicket-Data</td> <td><a href="#">troubleTicketData</a></td> <td>1..1</td> <td></td> </tr> <tr> <td>appointmentIdOld</td> <td>xs:long(totalDigits:10)</td> <td>1..1</td> <td></td> </tr> </tbody> </table>	Element	Type	Occ	Comment	troubleTicket-Data	<a href="#">troubleTicketData</a>	1..1		appointmentIdOld	xs:long(totalDigits:10)	1..1		1..1	
Element	Type	Occ	Comment												
troubleTicket-Data	<a href="#">troubleTicketData</a>	1..1													
appointmentIdOld	xs:long(totalDigits:10)	1..1													

### 3.3.1.9 bookAppointmentRequestType

**Description:** Purpose: Book a time slot for an appointment.

**Used by:** [bookAppointment](#)

**Extension of:** [requestType](#)

Element	Type	Occ	Comment
bookingSessionId	xs:string(maxLength:30)	1..1	The session id for the appointment booking (stateful in WSG and backend)
timeSlot	<a href="#">timeSlot</a>	1..1	A bookable time slot for an appointment
outOfSaSlaReason	xs:string(maxLength:20)	0..1	Reason, why the chosen time slot is out of sa sla. This field, that is required if the chosen timeSlot is out of SA_SLA (if outOfSaSla=true)

### 3.3.1.10 rescheduleAppointmentRequestType

**Description:** Purpose: Update the time and date of an appointment (reschedule). Books a new appointment and cancels the old one at the same time.

**Used by:** [rescheduleAppointment](#)

**Extension of:** [requestType](#)

Element	Type	Occ	Comment
bookingSessionId	xs:string(maxLength:30)	1..1	The session id for the appointment booking (stateful in WSG and backend)
timeSlot	<a href="#">timeSlot</a>	1..1	A bookable time slot for an appointment
outOfSaSlaReason	xs:string(maxLength:20)	0..1	Reason, why the chosen time slot is out of sa sla. This field, that is required if the chosen

			timeSlot is out of SA_SLA (if outOfSaSla=true)
appointmentIdOld	xs:long(totalDigits:10)	1..1	The id of an appointment to be replaced

### 3.3.1.11 cancelAppointmentRequestType

**Description:** Purpose: Cancel an appointment.

**Used by:** [cancelAppointment](#)

**Extension of:** [requestType](#)

Element	Type	Occ	Comment
appointmentId	xs:long(totalDigits:10)	1..1	The id of an appointment

### 3.3.1.12 requestType

**Description:** The abstract generic request message type.

**Extended by:** [getAvailableTimeSlotsRequestType](#), [bookAppointmentRequestType](#), [rescheduleAppointmentRequestType](#), [cancelAppointmentRequestType](#)

**Extension of:** [messageType](#)

Element	Type	Occ	Comment
ispId	xs:int(totalDigits:6)	1..1	The ISP/PTS id

### 3.3.1.13 getAvailableTimeSlotsResponseType

**Description:** Purpose: Get the available timeslots for an appointment.

**Used by:** [getAvailableTimeSlotsResponse](#)

**Extension of:** [ackType](#)

Element	Type	Occ	Comment
bookingSessionId	xs:string(maxLength:30)	0..1	The session id for the appointment booking (stateful in WSG and backend)
timeSlot	<a href="#">timeSlot</a>	0..n	A bookable time slot for an appointment

### 3.3.1.14 bookAppointmentResponseType

**Description:** Purpose: Response to appointment booking or moving (scheschedule) request.

**Used by:** [bookAppointmentResponse](#)

**Extension of:** [ackType](#)

Element	Type	Occ	Comment
appointment	<a href="#">appointment</a>	0..1	The object appointment

### 3.3.1.15 cancelAppointmentResponseType

**Description:** Purpose: Response to appointment cancel request.

**Used by:** [cancelAppointmentResponse](#)

**Extension of:** [ackType](#)

### 3.3.1.16 ackType

**Description:** The generic acknowledge response type.

**Extended**

**by:** [getAvailableTimeSlotsResponseType](#), [bookAppointmentResponseType](#), [cancelAppointmentResponseType](#)

**Extension of:** [responseType](#)

### 3.3.1.17 responseType

**Description:** The generic response message type.

**Extended by:** [ackType](#)

**Extension of:** [messageType](#)

Element	Type	Occ	Comment
success	xs:boolean	1..1	The flag indicating success or failure: True if request succeeded, false if not.
reason	xs:string(pattern:[\dA-Z]{3})	0..1	A 3-letter error code (aka messageId) where "000" means "ok".
reason-Comment	xs:string(minLength:1,maxLength:256)	0..1	Some additional textual description for the reason.
actionCode	xs:string(Enumeration: GET_TIMESLOTS,BOOK_APPOINTMENT,RESCHEDULE_APPOINTMENT,CANCEL_APPOINTMENT)	1..1	The action codes

### 3.3.1.18 messageType

**Description:** The abstract generic message type.

**Extended by:** [requestType](#), [responseType](#)

Attribute	Type	Use	Comment
correlationId	xs:string(maxLength:64)	optional	A unique ID to correlate request and answer(s) within a (asynch.) batch process.

### 3.3.1.19 troubleTicketData

**Description:** The fields used for appointment creation

**Used by:** [getAvailableTimeSlotsRequestType](#)

Element	Type	Occ	Comment
errorCategory	xs:int(totalDigits:3)	1..1	An error category LOV:={500:="xDSL Trouble Ticket", 501:="xDSL Installation Ticket", 502:="VoBB Trouble Ticket", 503:="VoBB Installation Ticket", 504:="Streaming Trouble Ticket", 505:="Streaming Installation Ticket", 507:="Splitter Installation Ticket", 508:="Streaming CPE TT", 509:="Additional Settop Box", 512:="IP-Centrex Trouble Ticket", 513:="IP-Centrex VoBB Business", 515:="Mobile Installation Ticket", 516:="VOBB (Business) Trouble Ticket", 517:="VOBB (Business) InstallationTicket", 541:="No Signal", 542:="No Connectionn", 543:="Interruptions", 544:="Performance Problem", 549:="Unkown", 550:="TV Internet down", 551:="No Signal", 552:="No Connectionn", 553:="Interruptions", 554:="Performance Problem", 555:="Freeze", 600:="FA Private"}
dnVnNsn	xs:string(pattern:0[1-9]\d{8})	1..1	A nomalized phone number, DN, VN or NSN (e.g. "0312223344")
eventBasedSla	xs:boolean	0..1	Equals to "true" if an eventBasedSla is chosen, "false" otherwise

### 3.3.1.20 timeSlot

**Description:** A bookable time slot for an appointment

**Used by:** [bookAppointmentRequestType](#), [rescheduleAppointmentRequestType](#), [getAvailableTimeSlotsResponseType](#)

Element	Type	Occ	Comment
Swisscom (Schweiz) AG CH-3050 Bern	Title Section Valid from Issue date		B2B Appointment Booking Specification <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b> <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b> 12nd May 2008
			Version <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b> Chapter <b>Fehler! Verweisquelle konnte nicht gefunden werden.</b>

timeSlotId	xs:int(totalDigits:6)	1..1	The timeslot id for an appointment
timeSlotSize	xs:int(totalDigits:3)	1..1	The size of the timeframe of the time slot. LOV:={1:="Whole day", 2:="Half day", 3:="Two hours"}
timeSlotStartDateTime	xs:dateTime	1..1	The start date and time of a time slot
timeSlotEndDateTime	xs:dateTime	1..1	The end date and time of a timeslot
outOfSaSla	xs:boolean	1..1	Flag, which says if the time slot is out of SA_SLA (equals to "true" if out of SA_SLA, "false" otherwise)
rank	xs:int(minInclusive:0,maxInclusive:100)	1..1	The rank of a time slot. Range:={0:="best", 100:="worst"}

### 3.3.1.21 appointment

**Description:** The object appointment

**Used by:** [bookAppointmentResponseType](#)

Element	Type	Occ	Comment
appointmentId	xs:long(totalDigits:10)	1..1	The id of an appointment
timeSlotStartDateTime	xs:dateTime	1..1	The start date and time of a time slot
timeSlotEndDateTime	xs:dateTime	1..1	The end date and time of a timeslot

## 4 Web Service Interface

The WSG Appointment Booking services are exposed through a Web service interface.

### 4.1 Security

The following sections describe the implemented precautions aiming to improve the security of data transport in terms of confidentiality and non-repudiation.

#### 4.1.1 Encryption

To meet the confidentiality aspect of information security the data transport between the web service client and its server is protected using the https schema.

Https is a URI scheme equivalent to the http scheme, originally intended to be used with the HTTP protocol, but with added encryption layer. The URI structure is the same, except that URIs begins with "https:" rather than "http:". The scheme was invented by Netscape Communications Corporation to provide authentication and encrypted communication and is widely used on the Web for security-sensitive communication, such as payment transactions.

Instead of using plain text socket communication, the session data is encrypted using either a version of the SSL (Secure Socket Layer) protocol or the TLS (Transport Layer Security) protocol, thus ensuring reasonable protection from eavesdroppers, and man in the middle attacks. The default TCP port of https: is 443.

#### 4.1.2 Authentication, Authorization, and Accounting (AAA)

To meet the accountability and non-repudiation aspects of information security each Web Service requires username tokens according to the OASIS Web Service Security (WS-Security) definition.

Please consult Appendix A for further details.

### 4.2 Model: WSDL and XML Schemas

The definition of the Web Services is defined in WSDL and based on the already described XML schema. Refer to the attached ZIP-file described above (section **Fehler! Verweisquelle konnte nicht gefunden werden.**).

### 4.3 The WSG Appointment Booking Outbound Web Service

The WSG Appointment Booking Outbound Web is published under the following URLs:

#### Version 1.0

Platform	URL	Description
Production	https://webservices.swisscom.com/wsg/prod/tt/WsgAppBookingV001	Production environment
ISP-Test	https://webservices.swisscom.com/wsg/isp/tt/WsgAppBookingV001	Test environment for PTS

### 4.3.1 Success or Error status

All response messages contain a member named “**success**”, which indicates if the operation has been successful or not and -- if not -- which error has been raised through its attributes “**reason**” (the error-code) and “**reasonComment**” (the error-text) (refer to the schema documentation for all details).

### 4.3.2 Error Codes

The WSG outbound web-services may return the following general error codes:

#### General Web Service Exceptions that may occur in all functions

ErrorCode / Reason	ErrorText / ReasonComment
618	Access Denied
G02	Bad Request
G01	System Error
G03	Request Timeout
G04	Backend Resource Temporarily Not Available
G05	Backend Resource Experiences Temporary Problems
G06	Backend Resource Reports Errors

#### Specific Exceptions for the function *getAvailableTimeSlots*

ErrorCode / Reason	ErrorText / ReasonComment
F29	Appointment Booking Is Not Allowed In Current Context

#### Specific Exceptions for the function *bookAppointment*

ErrorCode / Reason	ErrorText / ReasonComment
F30	Session Expired

#### Specific Exceptions for the function *rescheduleAppointment*

ErrorCode / Reason	ErrorText / ReasonComment
F30	Session Expired
F31	Invalid Appointment Id

#### Specific Exceptions for the function *cancelAppointment*

ErrorCode / Reason	ErrorText / ReasonComment
F31	Invalid Appointment Id

#### 4.4 Connection Testing

To test the connection you may access the web-service base URL with the ending query-parameter “?wsdl”:

https://<web-service-URL>**?wsdl**

The following facts can be stated if the operation returns successfully the WSDL:

- The WSG FA Outbound Web service is up and running
- The client could successfully connect to the Web service , which implies
  - The connection URL is correct
  - The call was successfully authenticated by PowerGate
  - The client session was considered valid by the Web service

**IMORTANT NOTE: Using this operation for keep-alive purposes (e.g. periodically pinging or likewise) is prohibited!**

## 5 Appendix A

### 5.1 Securing a WEB-Service with Powergate

A WEB-Service can be secured by using WSS Security (ref. OASIS Standard <http://www.oasis-open.org/specs/index.php#wssv1.0>). WSS Security Tokens have to be included in the header of the SOAP Requests. Realizing this feature is dependent of the implementation of the WS Clients. The header has to look as follows:

```
<soapenv:Header>
  <wsse:Security
    soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
    soapenv:mustUnderstand="0"
    xmlns:wsse=" http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken
      <wsse:Username>586221</wsse:Username>
      <wsse:Password>te3LsD43</wsse:Password>
    </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
```

Concerning "Username" and "Password" please refer to 5.5 Login .

### 5.2 Setting up a new WEB-Service

1. Determining the URL for the WEB-Service
2. Implementing the client in such a way that the WSS Security Tokens are included in the SOAP (see following example).

### 5.3 Sample Client with wss4j

wss4j will be used for creating the WSS Header (ref. <http://ws.apache.org/wss4j/>). The called service offers the method "list()" which will return the received header as a string.

```
public String[] doCall(
    com.swisscom.powergate.ws.TestwsSoapBindingStub binding)
    throws Exception {

    String userName = "58622141964";
    String pwd = "tXVLsD43";

    Stub bindingStub = (Stub) binding;

    Document doc = DocumentBuilderFactory.newInstance()
        .newDocumentBuilder().newDocument();

    Element element = doc.createElementNS(WSSConstants.WSSE_NS_OASIS_1_0,
        "wsse:" + WSSConstants.USERNAME_TOKEN_LN);
```



```

WSSecurityUtil.setNamespace(element, WSConstants.WSSE_NS_OASIS_1_0,
    WSConstants.WSSE_PREFIX);

// create username element
Element elementUsername = doc.createElementNS(
    WSConstants.WSSE_NS_OASIS_1_0, "wsse:"
        + WSConstants.USERNAME_LN);
WSSecurityUtil.setNamespace(elementUsername,
    WSConstants.WSSE_NS_OASIS_1_0, WSConstants.WSSE_PREFIX);
elementUsername.appendChild(doc.createTextNode(userName));
element.appendChild(elementUsername);

// create password element
Element elementPassword = doc.createElementNS(
    WSConstants.WSSE_NS_OASIS_1_0, "wsse:"
        + WSConstants.PASSWORD_LN);
WSSecurityUtil.setNamespace(elementPassword,
    WSConstants.WSSE_NS_OASIS_1_0, WSConstants.WSSE_PREFIX);
elementPassword.appendChild(doc.createTextNode(pwd));
element.appendChild(elementPassword);

// set the header
bindingStub.setHeader(WSConstants.WSSE_NS_OASIS_1_0, "wsse:Security",
    element);
assertNotNull("binding is null", binding);

// Time out after a minute
binding.setTimeout(60000);

return binding.list();
}

public void testHeaders() throws Exception {

    com.swisscom.powergate.ws.TestwsSoapBindingStub binding;
    try {

        binding = (com.swisscom.powergate.ws.TestwsSoapBindingStub) new
com.swisscom.powergate.ws.TestWSServiceLocator()
            .gettestws(new URL(
                "https://www.zugang.ch:44300/bg/services/testws"));

        Stub bindingStub = (Stub) binding;

        // keep session
        bindingStub.setMaintainSession(true);

        System.out.println("\n\rCALL 1");
        String[] headers = doCall(binding);
        for (int i = 0; i < headers.length; i++) {
            System.out.println(headers[i]);
        }

        System.out.println("\n\rCALL 2");
        headers = doCall(binding);
        for (int i = 0; i < headers.length; i++) {
            System.out.println(headers[i]);
        }

    } catch (javax.xml.rpc.ServiceException jre) {
        if (jre.getLinkedCause() != null)
            jre.getLinkedCause().printStackTrace();
        throw new junit.framework.AssertionFailedError(
            "JAX-RPC ServiceException caught: " + jre);
    }
    assertNotNull("binding is null", binding);
}

```



```
// Time out after a minute  
binding.setTimeout(60000);  
}
```

## 5.4 Errors

### Authentication error

In case of an authentication error the system will return *HTTP Status 403 Forbidden*. The error will be returned inside the SOAP body as <SOAP-ENV:Fault>

```
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Client</faultcode>
  <faultstring>Authentication required (realm='soap')</faultstring>
</SOAP-ENV:Fault>
```

### Backend Server not available:

```
HTTP/1.1 502 Bad Gateway
Date: Thu, 20 Apr 2006 11:49:22 GMT
Server: Apache
Pragma: no-cache
Connection: close
Cache-Control: no-cache
Content-Type: text/xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Upstream server is not available</faultstring>
      <faultactor>https://wstest.swisscom.com/wsg/omsol/bb/WsgBb</faultactor>
      <detail>Upstream server is not available</detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Error on the Backend Server

Here an example in the form of a standard http error:

```
HTTP/1.1 500 Internal Server Error
Date: Wed, 19 Apr 2006 14:27:03 GMT
Server: Apache
Set-Cookie: Navajo=AUCaUVoasEVlDVm29EUVyRYuqxwWRG4ozwlKJEewSpvKoVwVEz9mjIclAEQ0goaIH3ZnB9g/RXA-;
path=/; secure; HttpOnly
Set-Cookie: JSESSIONID=AEC47F88F1B35E4329C5C58F40841B4E; path=/wsg/e2e/bb/wsg-outbound; secure;
HttpOnly
Content-Type: text/xml; charset=utf-8
Connection: close
```

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode xmlns:ns1="http://xml.apache.org/axis/">
        ns1:Client
      </faultcode>
      <faultstring>No such operation 'list'</faultstring>
      <detail>
        <ns2:hostname xmlns:ns2="http://xml.apache.org/axis/">
          sbe18304.swissptt.ch
        </ns2:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

**Swisscom (Schweiz) AG**  
CH-3050 Bern

Title	B2B Appointment Booking Specification
Section	<b>Fehler! Verweisquelle konnte nicht gefunden werden.</b>
Valid from	<b>Fehler! Verweisquelle konnte nicht gefunden werden.</b>
Issue date	12nd May 2008

Version **Fehler! Verweisquelle konnte nicht gefunden werden.**  
Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**

```

                </ns2:hostname>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>

```

## Wrong URL (Webservice not existing)

```

HTTP/1.1 404 Not Found
Date: Wed, 19 Apr 2006 14:30:38 GMT
Server: Apache
Pragma: no-cache
Connection: close
Cache-Control: no-cache
Content-Type: text/xml

```

```

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>mapping for request URI '/wsgdf/e2e/bb/WsgBb' not found</faultstring>
      <faultactor>https://wstest.swisscom.com/wsgdf/e2e/bb/WsgBb</faultactor>
      <detail>mapping for request URI '/wsgdf/e2e/bb/WsgBb' not found</detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## No authorisation for the service but valid Login

```

HTTP/1.1 403 Forbidden
Date: Wed, 19 Apr 2006 14:31:40 GMT
Server: Apache
Pragma: no-cache
Connection: close
Cache-Control: no-cache
Content-Type: text/xml

```

```

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Your are not authorized to access the requested resource</faultstring>
      <faultactor>https://wstest.swisscom.com/wsg/prod/bb/WsgBb</faultactor>
      <detail>Your are not authorized to access the requested resource</detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 5.5 Login from the user point of view

Two possibilities can be used for Login:

- Login with **PUI** (Personal User Identification: 11-digit number) and **Password** (4 – 15 characters). The PIU will be sent to the user by email or letter after registration. It has to be used for the first Login. The password will be sent to the user by letter and can be changed arbitrarily by the user after the first Login.
- Login with username unique alphanumeric identification) and **Password** (4 – 15 digits). The username (formerly also known as synonym) van be created by the user after the first login with the



PUI. It must start with an alphabetic character, must be 7-20 characters in length and must be unique. Maybe the user needs several trials till a unique and not yet used name could be found. The password will be sent to the user by letter and can be changed arbitrarily by the user after the first Login.