

Broadband Connectivity Service (BBCS), Full Access Services (FA) and Cooperation Partner Services (COPA) General Functions Interface Specification

Version	29
Issue date	18.08.2017
Replaces version	28 or previous
Valid from	12.09.2017
Valid until	recalled or replaced by new version
Classification	Technical Support Documentation (BBCS and FA)
Status	released

Contents

Checklist of changes	3
Release	3
1 Introduction.....	5
1.1 Purpose.....	5
1.2 Target readership, requirements of the reader.....	5
1.3 Referenced documents.....	5
2 Web Service General.....	6
2.1 Security.....	6
2.1.1 Encryption.....	6
2.1.2 Authentication, Authorization, and Accounting (AAA).....	6
2.2 Success or Error status.....	7
2.2.1 Error Codes.....	7
2.3 Connection Testing.....	7
3 Notification Web-Service.....	8
3.1 Interface Definition.....	8
3.1.1 Latest Version.....	8
3.1.2 Deprecated Versions.....	9
4 COPA Notification Web-Service.....	9
4.1 Interface Definition.....	9
4.1.1 Latest Version.....	9
4.1.2 Deprecated Versions.....	11
5 Appointment-Booking Web-Service.....	12
5.1 Interface Definition.....	12
5.1.1 Latest Version.....	12
5.1.2 Deprecated Versions.....	12
6 Direct Entry to WSG Assurance GUI.....	13
7 Appendix A.....	14
7.1 Securing a Web-Service with Powergate.....	14
7.2 Setting up a new Web-Service.....	14
7.2.1 Sample Client with wss4j.....	14
7.3 Errors.....	16
7.3.1 Authentication error.....	16
7.3.2 Backend Server not available:.....	16
7.3.3 Error on the Backend Server.....	16
7.3.4 Wrong URL (Webservice not existing).....	17
7.3.5 No authorisation for the service but valid Login.....	17
7.4 Login from the user point of view.....	18

Checklist of changes

Version	Date	Changed by	Comments / nature of the change
14	15.09.2010	HP	Version WSG-8.2 / Nov10 - WSG V8.2 changes - TT Schema update - Integration of the "Appointment Booking" services
15	27.04.2011	HP	Version WSG-8.4 / Jun11 - Modifications in notification interface
16	27.04.2011	Roland Staub	Version WSG-9.4 / Mai13 - Modifications in notification interface
17	15.07.2013	Roland Staub	Version WSG-9.5 / Aug13 - Modifications in notification interface
18	14.10.2013	Roland Staub	Version WSG-9.6 / Nov13
19	06.02.2014	Roland Staub	Version WSG-9.7 / Feb14
20	25.04.2014	Roland Staub	Version WSG-9.8 / Mai14
21	08.08.2014	Roland Staub	Version WSG-9.9 / Aug14
22	16.10.2014	Roland Staub	Version WSG-9.10 / Nov14
23	13.02.2015	Roland Staub	Version WSG-9.11 / Mar15
24	20.05.2015	Roland Staub	Version WSG-9.12 / June15
24	29.10.2015	Roland Staub	Added Classification
25	29.04.2016	Roland Staub	Version WSG-9.16 / Mai16
26	29.10.2016	Roland Staub	Version WSG-9.18 / Nov16
27	14.03.2017	Roland Staub	Version WSG-9.19 / Mar17
28	18.05.2017	Roland Staub	Version WSG 17.06 / June17
29	18.08.2017	Roland Staub	Version WSG 17.09 / Sept17

Release

int.Version	Date	Released by	Comments / nature of the change
10	11.12.2008	SCS Peter Rosenberger	Released for internal use
11	26.02.2009	SCS Rolf Seltmann	Released for contract version 13-2
12	26.10.2009	SCS Rolf Seltmann	Released for contract version 13-31
13	27.04.2010	SCS Rolf Seltmann	Released for contract version 13-4
14	14.10.2010	SCS H. Künzi	Released for WSG-8.2 (Nov 2010)

15	03.05.2011	SCS H. Künzi	Released for WSG-8.4 (Jun 2011)
16	08.05.2013	SCS H.Künzi	Released for WSG-9.4 (Mai 2013)
17	16.07.2013	SCS H.Künzi	Released for WSG-9.5 (Aug 2013)
18	28.10.2013	SCS H.Künzi	Released for WSG-9.6 (Nov2013)
19	10.02.2014	SCS H.Künzi	Released for WSG-9.7 (Feb 2014)
20	02.05.2014	SCS H.Künzi	Released for WSG-9.8 (Mai2014)
21	08.08.2014	SCS H.Künzi	Released for WSG-9.9 (Aug2014)
22	30.10.2014	SCS H.Künzi	Released for WSG-9.10 (Nov2014)
23	17.02.2015	SCS H.Künzi	Released for WSG-9.11 (Mar2015)
24	27.05.2015	SCS H.Künzi	Released for WSG-9.12 (Jun2015)
25	04.05.2016	SCS H.Künzi	Released for WSG-9.16 (Mai2016)
26	01.11.2016	SCS H.Künzi	Released for WSG-9.18 (Nov2016)
27	15.03.2017	SCS H.Künzi	Released for WSG-9.19 (Mar2017)
28	22.05.2017	SCS H.Künzi	Released for WSG1706 (June2017)
29	25.08.2017	SCS H.Künzi	Released for WSG1709 (Sep2017)

1 Introduction

Besides the fully-fledged business processes there are some services that are used for different purposes and therefore cannot be assigned to a specific WSG feature.

1.1 Purpose

This document provides the detailed technical specification of the implementation of the following general usable services:

- Notifications
- Appointment Booking
- Direct Entry of WSG Assurance GUI

1.2 Target readership, requirements of the reader

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

1.3 Referenced documents

- [1] WSG Messages
- [2] PowerGate Integration Documentation

2 Web Service General

The synchronous B2B interfaces are realized as "Web-Services". A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as WSDL. Other systems interact with the Web service in a manner prescribed by its interface using messages, which may be enclosed in a SOAP envelope. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards. OASIS and the W3C are the primary committees responsible for the architecture and standardization of web services. To improve interoperability between web service implementations, the WS-I organization has been developing a series of profiles to further define the standards involved.

2.1 Security

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

2.1.1 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.

2.1.2 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 the PowerGate integration documentation [1] for further details.

2.2 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).

2.2.1 Error Codes

The WSG outbound web-services may return the following general error codes (please refer to document [1] for a list of all error codes / messages):

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

2.3 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 tested WSG outbound web service is up and running
- The client could successfully connect to the web service , which implies
 - o The connection URL is correct
 - o The call was successfully authenticated by PowerGate
 - o The client session was considered valid by the web service

IMPORTANT NOTE:

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

3 Notification Web-Service

The **Notification** web service offers information about the progress of the processing of the different user requests (orders, tickets, information requests) that exist in WSG. The notifications are created during the request processing according to the ISP/PTS specific configuration and stored in the system for a defined period. During this period the notifications can be retrieved by the ISP/PTS.

3.1 Interface Definition

3.1.1 Latest Version

The current Interface-Descriptions are provided based on a XML-File definition given by the following schemas:

wsgNotification_v20.0.1.zip

The ZIP-File contains:

ChangeLog.txt	The details of the changes per released version (change-history)
doc/*	HTML based documentation of the interface (model, messages, types, ...)
schema/	The interface definition (WSDL, Schema)

How to use samples can be found in the B2B_Web_Service_Tutorial.doc

The following table provides an overview of the available business processes for “Notifications”:

Business Process	Request Type	Response Type
Get Notification	getNotificationRequestType	getNotificationResponseType

The “Get Notification” can be requested for the following types:

- BB Orders: bbcsOrder
- (Onsite) Installation Tickets: osInstallationTicket
- BB Trouble Tickets: bbcsAccessTt
- BB Connectivity Tickets: bbcsConnectivityTt
- FA/TAL Orders: faOrder
- FA/TAL Trouble Tickets: faTt
- FA/TAL Clarification Tickets: faClarificationT
- Information Requests: informationRequest
- Voice Messages (related to TDM connection): voiceMessage
- ChangeTickets: changeTicket
- networkCustomerNotification
- aloOrder
- aloTt

The WSG Notification Outbound Web is published under the following URLs:

Platform	URL	Description
Production	https://webservices.swisscom.com/wsg/prod/bb/WsgNotificationV020 https://webservices.swisscom.com/wsg/prod/bb/WsgNotificationV019 https://webservices.swisscom.com/wsg/prod/bb/WsgNotificationV018 https://webservices.swisscom.com/wsg/prod/notif/WsgNotificationV020 https://webservices.swisscom.com/wsg/prod/notif/WsgNotificationV019 https://webservices.swisscom.com/wsg/prod/notif/WsgNotificationV018	Production environment, actual version Production environment, previous version
ISP-Test	https://webservices.swisscom.com/wsg/isp/bb/WsgNotificationV020 https://webservices.swisscom.com/wsg/isp/bb/WsgNotificationV019 https://webservices.swisscom.com/wsg/isp/bb/WsgNotificationV018 https://webservices.swisscom.com/wsg/isp/notif/WsgNotificationV020 https://webservices.swisscom.com/wsg/isp/notif/WsgNotificationV019 https://webservices.swisscom.com/wsg/isp/notif/WsgNotificationV018	ISP/PTS Test environment, actual version ISP/PTS Test environment, previous version

3.1.2 Deprecated Versions

With this software release a new version of the Assurance B2B specification is provided (according the table above); all new development activities shall use this version.

The previous version (version number one level lower than the new one) remains valid, but will become deprecated as soon as a possible next version of this interface will be created and published.

4 COPA Notification Web-Service

The **CopaNotification** web service offers information about the progress of the processing of the different user requests (orders, tickets, information requests) that exist in WSG. The notifications are created during the request processing according to the COPA specific configuration and stored in the system for a defined period. During this period the notifications can be retrieved by the COPA.

4.1 Interface Definition

4.1.1 Latest Version

The current Interface-Descriptions are provided based on a XML-File definition given by the following schemas:
wsgCopaNotification_v4.0.1.zip

The ZIP-File contains:

ChangeLog.txt	The details of the changes per released version (change-history)
doc/*	HTML based documentation of the interface (model, messages, types, ...)
schema/	The interface definition (WSDL, Schema)

How to use samples can be found in the B2B_Web_Service_Tutorial.doc

The following table provides an overview of the available business processes for “Notifications”:

Get Notification	getNotificationRequestType	getNotificationResponseType

The “Get Notification” can be requested for the following types:

- Information Requests: informationRequest
- Copa Orders: copaOrder
- Copa Tt: copaTt

The WSG COPA Notification Outbound Web is published under the following URLs:

Platform	URL	Description
Production	https://webservices.swisscom.com/wsg/prod/copa/notif/WsgCopaNotificationV04 https://webservices.swisscom.com/wsg/prod/copa/notif/WsgCopaNotificationV03	Production environment, actual version
ISP-Test	https://webservices.swisscom.com/wsg/isp/copa/notif/WsgCopaNotificationV04 https://webservices.swisscom.com/wsg/isp/copa/notif/WsgCopaNotificationV03	COPA Test environment, actual version

4.1.2 Deprecated Versions

With this software release a new version of the Assurance B2B specification is provided (according the table above); all new development activities shall use this version.

The previous version (version number one level lower than the new one) remains valid, but will become deprecated as soon as a possible next version of this interface will be created and published.

5 Appointment-Booking Web-Service

The **Appointment Booking** web service offers the reservation of time slots for the Swisscom field force organisation and has to be used in several business processes.

5.1 Interface Definition

5.1.1 Latest Version

The current Interface-Descriptions are provided based on a XML-File definition given by the following schemas:
wsgAppBooking_v3.0.1.zip

The ZIP-File contains:

ChangeLog.txt	The details of the changes per released version (change-history)
doc/*	HTML based documentation of the interface (model, messages, types, ...)
schema/	The interface definition (WSDL, Schema)

How to use samples can be found in the B2B_Web_Service_Tutorial.doc

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

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

Platform	URL	Description
Production	https://webservices.swisscom.com/wsg/prod/tt/WsgAppBookingV003	Production environment, actual version
ISP-Test	https://webservices.swisscom.com/wsg/isp/tt/WsgAppBookingV003	ISP/PTS Test environment, actual version

5.1.2 Deprecated Versions

At the moment no previous (deprecated) versions of this web service exist.

6 Direct Entry to WSG Assurance GUI

The following table provides an overview of the available direct entry points for the **Assurance GUI** of WSG (for "OneCRM"):

Business Process	Function	Parameters
Get TT-Detail	TtDetail	TT_ID (WSG Trouble Ticket Id)

The following URLs have to be used for this function:

Platform	URL	Description
Production	https://wholesale.swisscom.com/powergate/wsgtt/tt/tt_b2b_entry.epl	Production environment
ISP-Test	https://wholesale.swisscom.com/powergate/wsgtttest/tt/tt_b2b_entry.epl	ISP Test environment

Example on the production platform (with automatic login mechanism on the client system):

https://wholesale.swisscom.com/wsg/powergate/wsgtt/tt/tt_b2b_entry.epl?function=TtDetail&TT_ID=123456

7 Appendix A

7.1 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 chapter 7.4 "Login from the user point of view".

7.2 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).

7.2.1 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(WsConstants.WSSE_NS_OASIS_1_0,
        "wsse:" + WsConstants.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);
}

```

7.3 Errors

7.3.1 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>

```

7.3.2 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/oms01/bb/wsgBb</faultactor>
      <detail>Upstream server is not available</detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

7.3.3 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=AUCaUVoasEVlDvm29EUvyRYuqxwWRG4ozw1KJEwSpvKoVwVEz9mjIclAEQ0goaIH3ZnB9g/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>

```

7.3.4 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>

```

7.3.5 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>

```

7.4 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 PUI 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) can 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.