



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für
Wirtschaft, Bildung und Forschung WBF

Staatssekretariat für Wirtschaft SECO

SOAP Schnittstelle von simap.ch

Version 3.12

Inhaltsverzeichnis

1	Einleitung	5
2	Nutzung der WEB-Services	6
2.1	Übersicht über alle verfügbaren Methoden.....	6
2.2	Genereller Zugriff auf die WEB-Services	8
2.3	Zugriff auf die WEB-Services mit Benutzerprofil.....	8
2.3.1	Methode getActualCantonNoticeList()	8
2.3.2	Methode getActualCantonNoticeListForCategory(category)	9
2.3.3	Methode getNoticeList(publishDate).....	9
2.4	Zugriff auf die WEB-Services ohne Benutzerprofil.....	10
2.4.1	getNoticeXml(documentId)	10
2.4.2	getNoticeHtml(documentId).....	10
2.4.3	getSearchNoticeList(searchXml)	10
2.4.4	getSearchNoticeCount(searchXml)	14
2.4.5	getSearchNoticeXml(searchXml, languageKey)	14
	Anhang	17
2.5	Liste der Publikationstypen	17
2.6	Fehlermeldungen	17
2.7	JAVA – Beispiel	18
2.8	Generierung von Client Klassen.....	19
2.8.1	Beispiel - Programm.....	19
2.9	Beispiel eines SOAP-Requests	20

Abkürzungsverzeichnis

Abkürzung	Beschrieb
HTTP	Hypertext Transfer Protocol
OSI	Open Systems Interconnection model
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Markup Language

Glossar

Bezeichnung	Beschrieb				
IP	Das Internet Protocol (IP) ist ein in Computernetzen weit verbreitetes Netzwerkprotokoll und stellt durch seine Funktion die Grundlage des Internets dar. Das IP ist die Implementierung der Internetschicht des TCP/IP-Modells bzw. der Vermittlungsschicht des ISO/OSI-Referenzmodells. IP ist ein verbindungsloses Protokoll.				
ISO/OSI-Referenzmodell	Das ISO/OSI-Referenzmodell (Open Systems Interconnection model) ist ein Referenzmodell für Netzwerkprotokolle als Schichtenarchitektur. Es wird seit 1983 von der International Telecommunication Union (ITU) und seit 1984 auch von der International Organization for Standardization (ISO) als Standard veröffentlicht.				
SOAP	SOAP (Simple Object Access Protocol) ist ein Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls (WEB-Services) durchgeführt werden können. SOAP ist ein industrieller Standard des World Wide Web Consortiums (W3C).				
TCP	Das Transmission Control Protocol (TCP) ist ein Netzwerkprotokoll, das definiert, auf welche Art und Weise Daten zwischen Netzwerkkomponenten ausgetauscht werden sollen. Nahezu sämtliche aktuelle Betriebssysteme moderner Computer beherrschen TCP und nutzen es für den Datenaustausch mit anderen Rechnern. Das Protokoll ist ein zuverlässiges, verbindungsorientiertes, paketvermitteltes Transportprotokoll in Computernetzwerken. Es ist Teil der Internetprotokollfamilie, der Grundlage des Internets.				
TCP/IP-Referenzmodell	Kommunikation wird in Rechnernetzen durch Netzwerkprotokolle umgesetzt und in der Praxis in funktionale Schichten (Layer) unterteilt. Für das Internet und die Internetprotokollfamilie ist dabei die Gliederung nach dem sogenannten TCP/IP-Referenzmodell, welches 4 aufeinander aufbauende Schichten beschreibt, massgebend: <table border="1" data-bbox="406 1951 1406 2002"> <thead> <tr> <th>TCP/IP-Schicht</th> <th>Beispiel</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	TCP/IP-Schicht	Beispiel		
TCP/IP-Schicht	Beispiel				

Bezeichnung	Beschrieb	
	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, DHCP, OPC UA, TLS, SOCKS
	Transport	TCP, UDP, SCTP
	Internet	IP (IPv4, IPv6), ICMP (über IP)
	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI
WEB-Services	Ein WEB-Service (auch Webdienst) stellt eine Schnittstelle für die Maschine-zu-Maschine- oder Anwendungs-Kommunikation über Rechnernetze wie das Internet zur Verfügung. Die Kommunikation kann über Internet-Protokolle wie HTTP oder HTTPS erfolgen. Über diese Protokolle können Daten beispielsweise im XML-Format übertragen werden. Ein WEB-Service ist plattformunabhängig und steht in der Regel mehreren Programmen zum Aufrufen bereit.	
XML	Die Extensible Markup Language (XML), ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei, die sowohl von Menschen als auch von Maschinen lesbar ist. XML wird auch für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt, insbesondere über das Internet. XML ist eine Metasprache, auf deren Basis durch strukturelle und inhaltliche Einschränkungen anwendungsspezifische Sprachen definiert werden. Diese Einschränkungen werden entweder durch eine Document Type Definition (DTD) oder durch ein XML Schema ausgedrückt. Beispiele für XML-Sprachen sind: RSS, MathML, GraphML, XHTML, XAML, Scalable Vector Graphics (SVG), GPX, aber auch das XML-Schema selbst.	
WSDL	Die Web Services Description Language (WSDL) ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Netzwerkdienste (WEB-Services) zum Austausch von Nachrichten auf Basis von XML. WSDL ist ein industrieller Standard des World Wide Web Consortiums (W3C).	

1 Einleitung

simap.ch bietet Umsystemen eine Schnittstelle zum Bezug von simap.ch Publikationsdaten an. Dies wird durch ein Application Programming Interface (API) mittels des Netzwerkprotokolls Simple Object Access Protocol (SOAP) ermöglicht. Dieses Dokument enthält alle Informationen, die die Nutzung der Schnittstelle beschreiben.

SOAP stützt sich auf XML zur Repräsentation der Daten und auf Internet-Protokolle der Transport- und Anwendungsschicht gemäss dem TCP/IP-Referenzmodell zur Übertragung von Nachrichten. Die gängigste Kombination ist SOAP über HTTP und TCP mit dem man WEB-Services bereitstellt. Mit WEB-Services kann man von einem anderen Programm, das in einer beliebigen Programmiersprache (z.B.: Python, Java, C#, PHP) geschrieben sein kann, auf die von simap.ch bereitgestellten Methoden (Funktionen) zugreifen.

Anmerkung: Für Aufrufe einiger Methoden ist eine Anmeldung erforderlich. Benutzer-Profile können auf der Plattform <https://www.simap.ch> erfasst werden.

2 Nutzung der WEB-Services

2.1 Übersicht über alle verfügbaren Methoden

#	Methoden-Name	Beschreibung	Ae ¹
1	getActualCantonNoticeList()	Die Methode gibt alle IDs von freigegebenen und publizierten Meldungen zurück, deren Redaktionsschluss in den letzten 14 Tagen erreicht wurde.	Ja
2	getActualCantonNoticeListForCategory(category)	Die Methode gibt alle IDs von freigegebenen und publizierten Meldungen zurück, deren Redaktionsschluss in den letzten 14 Tagen erreicht wurde.	Ja
3	getNoticeList(publishDate)	Eine kantonale Druckerei erhält alle Dokument IDs von Meldungen mit dem angegebenen Publikationsdatum. Es werden nur Meldungen übermittelt, bei denen der Status der Meldung „Freigegeben“ oder „Publiziert“ ist.	Ja
4	getNoticeXml(documentId)	Durch die Methode erhält man die XML-Beschreibung der Meldung mit der angegebenen Meldungs-ID. Die XML-Beschreibung der Meldung kann über https://www.simap.ch/shab-forms/doc/dtd/SimapDtd.zip bezogen werden.	Nein
5	getNoticeHtml(documentId)	Durch die Methode erhält man die HTML-Beschreibung der Meldung mit der angegebenen Meldungs-ID. Die HTML-Beschreibung der Meldung kann über https://www.simap.ch/shab-forms/doc/dtd/SimapDtd.zip bezogen werden.	Nein
6	getSearchNoticeList(searchXml)	Die Methode „getSearchNoticeList“ führt eine Recherche in simap.ch durch. Die Methode gibt eine Liste aller Meldungsnummern zurück. Mit Hilfe der Meldungsnummern kann	Nein

¹ Ae = Anmeldung erforderlich?

Ja = Diese Methoden können nur von Benutzern mit den Benutzerprofilen «Kantonaler Administrator» oder «Kantonale Druckerei» verwendet werden.

Nein = Diese Methoden benötigen kein Benutzerprofil und können von allen interessierten Personen verwendet werden.

#	Methoden-Name	Beschreibung	Ae ¹
		eine Übersicht in Zusammenhang mit der Methode „getNoticeXml(documentId)“ erzeugt werden.	
7	getSearchNoticeCount(searchXml)	Die Methode „getSearchNoticeCount“ führt eine Recherche in simap.ch durch. Die Methode gibt die Anzahl aller Meldungen zurück. Der Aufruf der Methode ist der Gleiche wie der von „getSearchNoticeList(searchXml)“. Als Rückgabewert erhält man im Gegensatz zu „getSearchNoticeList(searchXml)“ die Anzahl aller Meldungen.	Nein
8	getSearchNoticeXml(searchXml, languageKey)	Die Methode „getSearchNoticeXml(searchXml, languageKey)“ führt eine Recherche für Publikationen in der angegebenen Sprache in simap.ch durch. Die Methode gibt eine Liste aller Meldungsnummern zurück. Mit Hilfe der Meldungsnummern kann eine Übersicht in Zusammenhang mit der Methode „getNoticeXml(documentId)“ erzeugt werden.	Nein
9	getAuthentication	Die Methode „getAuthentication“ ist zum Testen der Benutzer-Passwort-Kombination da und gibt im Erfolgsfall „true“ zurück.	Nein
10	getFaultInfo.getCode()	Im Falle eines Fehlers wird der Fehlercode durch diese Methode zurückgeliefert.	Nein
11	getFaultInfo.getMessage()	Im Falle eines Fehlers wird durch diese Methode die Fehlerbeschreibung zurückgeliefert.	Nein

2.2 Genereller Zugriff auf die WEB-Services

Der Zugriff auf den Soap-Service erfolgt über folgende URL: <https://www.simap.ch/soapserver>.

2.3 Zugriff auf die WEB-Services mit Benutzerprofil

Der Zugriff auf die Methoden 1 bis 3 gemäss Kapitel 2.1, Übersicht über alle verfügbaren Methoden, benötigen eine Authentifizierung und stehen nur den kantonalen Druckereien zur Verfügung. Es ist zwingend notwendig, dass die Authentifizierung im HTTPS-Header durchgeführt wird.

Die Methoden 4 bis 11 gemäss Kapitel 2.1, Übersicht über alle verfügbaren Methoden, stehen selbstverständlich auch Personen mit Benutzerprofil zur Verfügung.

Falls die Authentifizierung nicht im HTTPS-Header für die angesprochene Methode mitgeschickt wird, wird eine Fehlermeldung zurückgegeben. Die Authentifizierung erfolgt mit Hilfe der beiden Parameter `username` (Benutzername) und `password` (Password) im HTTPS-Header. Der HTTPS-Header sieht folgendermaßen aus:

```
POST /soapserver HTTP/1.1
username: [BENUTZERNAME]
SOAPAction: ""
Accept: text/xml, multipart/related, text/html, image/gif, image/jpeg, *;
q=.2, */*; q=.2
Content-Type: text/xml; charset=utf-8
password: [PASSWORT]
User-Agent: Java/1.6.0_23
Host: www.simap.ch
Connection: keep-alive
Content-Length: 273
```

Der Benutzername und das Password sind dieselben, die auch zum Anmelden in `simap.ch` verwendet werden. Die Authentifizierung kann durch die Methode `getAuthentication` überprüft werden.

2.3.1 Methode `getActualCantonNoticeList()`

Argumente: Keine

Beschreibung: Die Methode gibt alle IDs von freigegebenen und publizierten Meldungen zurück, deren Redaktionsschluss in den letzten 14 Tagen erreicht wurde. Es werden nur Meldungen übermittelt, bei denen der Status der Meldung „Freigegeben“ oder „Publiziert“ ist.

Verfügbarkeit: Die Methode ist nur verfügbar, wenn der Benutzer die Rolle „Kantonaler Administrator“ oder „Kantonale Druckerei“ hat.

2.3.2 Methode `getActualCantonNoticeListForCategory(category)`

Argumente:

- *category* Rubrikkürzel, das aus zwei Zeichen besteht.
Möglicher Wert: „OB“=Öffentliche Beschaffungen

Beschreibung: Die Methode gibt alle IDs von freigegebenen und publizierten Meldungen zurück, deren Redaktionsschluss in den letzten 14 Tagen erreicht wurde. Es werden nur Meldungen übermittelt, bei denen

- der Status der Meldung „Freigegeben“ oder „Publiziert“ ist.
- In `simap.ch` kann [*category*] nur den Wert „OB“ haben.

Verfügbarkeit: Die Methode ist nur verfügbar, wenn der Benutzer die Rolle „kantonaler Administrator“ oder „Kantonale Druckerei“ hat.

2.3.3 Methode `getNoticeList(publishDate)`

Argumente:

- *publishDate* Ein Publikationsdatum (Format: tt.mm.jjjj)

Beschreibung: Eine kantonale Druckerei erhält alle Dokument IDs von Meldungen mit dem angegebenen Publikationsdatum. Es werden nur Meldungen übermittelt, bei denen der Status der Meldung „Freigegeben“ oder „Publiziert“ ist.

Verfügbarkeit: Die Methode ist nur verfügbar, wenn der Benutzer die Rolle „Kantonaler Administrator“ oder „Kantonale Druckerei“ hat.

2.4 Zugriff auf die WEB-Services ohne Benutzerprofil

Im Folgenden werden alle WEB-Service-Methoden, die ohne Anmeldung verfügbar sind, beschrieben. In diesem Fall muss der HTTPS Header nicht geändert/erweitert werden, wie dies Kapitel 2.3 Zugriff auf die WEB-Services mit Benutzerprofil der Fall ist.

2.4.1 getNoticeXml(documentId)

Argumente:

- *documentId* Eine Meldungsnummer

Beschreibung: Durch die Methode erhält man die XML-Beschreibung der Meldung mit der angegebenen Meldungs-ID. Die XML-Beschreibung der Meldung kann über <https://www.simap.ch/shabforms/doc/dtd/SimapDtd.zip> bezogen werden.

Verfügbarkeit: Die Methode ist ohne Anmeldung für publizierte Meldungen verfügbar. Falls der Benutzer als kantonaler Administrator oder kantonale Druckerei angemeldet ist, werden auch Meldungen angezeigt, die den Status „Freigegeben“ haben.

2.4.2 getNoticeHtml(documentId)

Argumente:

- *documentId* Eine Meldungsnummer

Beschreibung: Durch die Methode erhält man die HTML-Beschreibung der Meldung mit der angegebenen Meldungs-ID. Die HTML-Beschreibung der Meldung kann über <https://www.simap.ch/shabforms/doc/dtd/SimapDtd.zip> bezogen werden.

Verfügbarkeit: Die Methode ist ohne Anmeldung verfügbar.

2.4.3 getSearchNoticeList(searchXml)

Argumente:

- *searchXml* besteht auf einer Liste von Feldnamen und Werten, nach denen gesucht werden soll. Erlaubte Feldnamen sind:

Bedeutung	Feldname	Werte
Stichworte	KEYWORDS	Stichwortsuche (Volltextsuche)
Ort der Auftragserfüllung	CONT_LOC_OB	Freitextfeld für den Ort
Zeitraum (fest)	TIMESPAN	TODAY (Heute), WEEK (Diese Woche), YEAR (Dieses Jahr)
Zeitraum (variabel)	STAT_TM_1 STAT_TM_2	Datum im Format DD.MM.JJJJ Werden beide Werte gefüllt wird im Intervall gesucht, wird nur STAT_TM_1 gefüllt wird nach dem gegebenen Tag gesucht.

Unterrubrik	TYPE_CD_OB	OB00,OB01,OB05,OB07 =Ausschreibungen OB02,OB08=Zuschläge OB03,OB04,OB06,OB09 = Andere Veröffentlichungen Siehe auch im Anhang : Liste der Publikationstypen
Laufende Verfahren	DEADLINE_PASSED_OB	N = Ja oder leer = Alle
Auftragsart	TYPE_CONTRACT_CD_OB	SUPPLIES (Lieferung) SERVICES (Dienstleistung) WORKS (Bauleistung) CONTEST (Wettbewerb) Mehrere Werte müssen kommasepariert werden
Verfahrensart	PROC_CD_OB	OPEN (Offen) RESTRICTED (Selektiv) OTHER (Freihändiges Verfahren)
CPV	CPV_OB	CPV Nummern kommasepariert
BKP	BKP_OB	BKP Nummern kommasepariert
Art der Auftraggeber	AUTH_CD_OB	FEDERATION (Bund (Zentrale Bundesverwaltung)), UTILITY (Bund (Dezentrale Bundesverwaltung - öffentlich rechtliche Organisationen), CANTON (Kanton), CANTON_OTHER (Andere Träger kantonaler Aufgaben), FOREIGN (Ausland), MUNICIPALITY (Gemeinde/Stadt), MUNICIPALITY_OTHER (Andere Träger kommunaler Aufgaben)
Kantone	KANTON_CD_OB	Kantonskürzel Beispiele: „ZH“ oder „ZH, FR“
Städte	CITY_CD_OB	SIMAP_CITY_[Stadtnummer] Mehrere Werte müssen kommasepariert werden.

		Erlaubte Stadtnummern siehe Städteliste Beispiel: < CITY_CD_OB> SIMAP_CITY_01< /CITY_CD_OB>
Meldungsnummer	NOTICE_NR	
Projekt-ID	PROJECT_ID_OB	

Zusätzlich enthält *searchXml* folgende Attribute:

PAGENO Nummer der anzuzeigenden Seite
RECORDPERPAGE Anzahl der Einträge pro Seite

Anmerkungen:

- Wenn einer der beiden Attribute mit -1 belegt wird, wird die gesamte Liste zurückgegeben.
- Es werden nie mehr als 1000 Ergebnisse zurückgeliefert.

Beispiel:


```
<search pageNo="1" recordsPerPage ="20" >
  <field name ="KEYWORD"><value>Tunnel</value></field>
  <field name ="TYPE_CONTRACT_CD_OB"><value>WORKS</value></field>
</search>
```

Ausgabe von Fehlern: Folgende Fehler werden ausgegeben, wenn die Recherche fehlschlägt:

Fehlername	Bedeutung
SEARCH_TOO_COMPLEX	Rechercheanfrage ist zu komplex und dauert daher zu lange
SEARCH_ERROR_WILDCARD_EXPANSION	Bei der trunkierten Suche gab es zu viele Treffer. Bitte grenzen Sie bei der Recherche denjenigen Suchbegriff weiter ein, der einen "*" Operator benutzt!
SEARCH_ERROR_COLUMN_NOT_INDEXED	Die Recherche kann derzeit wegen Wartungsarbeiten nicht durchgeführt werden. Versuchen Sie es zu einem späteren Zeitpunkt erneut.
SEARCH_ERROR_SYNTAX	Die Recherche ist syntaktisch nicht korrekt. Bitte verbessern Sie die Recherche.
SEARCH_ERROR_NUMERIC_OR_VALUE	Es wurde kein numerischer Wert übergeben
SEARCH_EMPTY	Rechercheanfrage ist leer
SEARCH_ERROR_20000	Das Publikationsdatum (vom) liegt ausserhalb dem erlaubten Suchbereich!
SEARCH_ERROR_20001	Das Publikationsdatum (bis) liegt ausserhalb dem erlaubten Suchbereich!

SEARCH_ERROR_20002	Die Suche ist zu unspezifisch!
SEARCH_ERROR_20003	Das gesuchte Wort ist ein so genanntes 'Stoppwort', das keinen Rückschluss auf den Dokumentinhalt zulässt. Nach Stoppwörtern kann deshalb nicht gesucht werden - bitte ändern Sie den verwendeten Suchbegriff.
SEARCH_ERROR_20004	Der verwendete Suchbegriff liefert zu viele Treffer - bitte ändern!
SEARCH_ERROR_20005	Der Server ist zur Zeit ausgelastet. Bitte versuchen Sie die Recherche zu einem späteren Zeitpunkt erneut!

Städteleiste:

Nummer	Wappen	Stadt
01		Bern
02		Biel
04		Lugano
05		Neuenburg
06		Winterthur
07		Zürich
08		St. Gallen
09		Lausanne

Beschreibung: Die Methode „getSearchNoticeList“ führt eine Recherche in simap.ch durch. Die Methode gibt eine Liste aller Meldungsnummern zurück. Mit Hilfe der Meldungsnummern kann eine Übersicht in Zusammenhang mit der Methode „getNoticeXml“ erzeugt werden.

WSDL Definition:

```
<message name="getSearchNoticeList">
  <part name="searchXml" type="xsd:String"/></message>
<message name="getSearchNoticeListResponse">
  <part name="noticeList" type="ns9:longArray"/>
</message>
```

Verfügbarkeit: Die Methode ist ohne Anmeldung für publizierte Meldungen verfügbar. Falls der Benutzer als „Kantonaler Administrator“ oder „Kantonale Druckerei“ angemeldet ist, werden auch Meldungen angezeigt, die den Status „Freigegeben“ haben.

2.4.4 getSearchNoticeCount(searchXml)

Argumente:

- *searchXml* analog Methode getSearchNoticeList(searchXml) gemäss Kapitel 2.4.3.

Beschreibung: Die Methode „getSearchNoticeCount“ führt eine Recherche in simap.ch durch. Die Methode gibt die Anzahl aller Meldungen zurück. Der Aufruf der Methode ist der Gleiche wie der von „getSearchNoticeList(searchXml)“. Lediglich die Rückgabe ist anders.

WSDL Definition:

```
<message name="getSearchNoticeCount">
  <part name="searchXml" type="xsd:String"/></message>
<message name=" getSearchNoticeListResponse">
  <part name="resultXml" type="xsd:String"/></message>
</message>
```

Folgender resultXml String wird zurückgeliefert:

```
<result>
  <notices>301</notices>
  <invitations>24</invitations>
  <awards>243</awards>
  <others>34</others>
</result>
```

Verfügbarkeit: Die Methode ist ohne Anmeldung für publizierte Meldungen verfügbar. Falls der Benutzer als „Kantonaler Administrator“ oder „Kantonale Druckerei“ angemeldet ist, werden auch Meldungen verrechnet, die den Status „Freigegeben“ haben.

2.4.5 getSearchNoticeXml(searchXml, languageKey)

Argumente:

- *searchXml* analog Methode getSearchNoticeList(searchXml) gemäss Kapitel 2.4.3.
- *languageKey* DE, FR, IT, EN

Beschreibung: Es wird eine Suche mit Hilfe des Arguments „searchXml“ definiert. Auf Basis dieser Suche wird ein XML mit einer Liste von publizierten Meldungen zurückgegeben. Für jede gefundene Meldung steht im zurückgegebenen XML ein Tag „item“ mit der Meldungsnummer. Für jede Meldung werden die Felder die weiter unten in „Auszeichnung einer Meldung“ beschrieben sind, ausgegeben.

Anmerkung: Die Liste der Felder ist nicht vollständig. Es werden nur die wichtigsten Felder zurückgegeben. Mit Hilfe der Methode „getNoticeXml“ werden alle Felder zurückgegeben.

WSDL Definition:

```
<message name="getSearchNoticeXml">
  <part name="searchXml" type="xsd:String"/></message>
  <part name="languageKey" type="xsd:String"/></message>
<message name=" getSearchNoticeListResponse">
  <part name="resultXml" type="xsd:String"/></message>
</message>
```

Beispiel der Rückgabe „resultXml“ für Meldungsnummer 812133

```
<?xml version="1.0" encoding="UTF-8" ?>
<itemList>
  <item id=812133>
    <projectid> 110143</projectid>
    <publicationDate>25.03.2014</ publicationDate >
    <deadline>24.03.2014 16:00</deadline>
    <type>Ausschreibungen</type>
    <contType>Bau</contType>
    <proc>Offenes Verfahren</proc>
    <description>Réalisation d'un stade de football et d'un
bâtiment de service CFC 211 - Lot 04 Gros oeuvre</descrip-
tion>
    <authName>Office des Bâtiments - selon dossier d'appels
d'offre K2</authName>
    <contLoc>Bern</contLoc>
    <cpv>CPV: 45223220, Baukostenplannummer </cpv>
    <bkp>(BKP): 211</bkp>
    <lotList>
      <lot>
        <number>...</number>
        <description>...</description>
        <cpv>...</cpv>
        <bkp>...</bkp>
        <npk>...</npk>
        <quantity>...</quantity>
      </lot>
    </lotList>
  </item>
  [...]
</itemList>
```

In diesem Beispiel sind alle zur Verfügung gestellten Auszeichnungen (Tags) enthalten.
In der Tabelle unten sind alle Auszeichnungen zusammengefasst:

Tabelle: Auszeichnungen einer Meldung

Element Name	Bedeutung
projectid	Projektnummer
publicationDate	Publikationsdatum
deadline	Eingabefrist
contType	Auftragsart
proc	Verfahrensart
description	Projektbeschreibung
authName	Beschaffungsstelle
contLoc	Ausführungsort
cpv	CPV-Code
bkp	BKP-Code
lotList	Liste von Losen (Wenn vorhanden)

lot.number	Losnummer
lot.description	Los Beschreibung
lot.cpv	CPV Codes des Loses
lot.bkp	BKP Codes des Loses
lot.npk	NPK Codes des Loses
lot.quantity	Umfang bzw. Menge des Loses

Verfügbarkeit: Die Methode ist ohne Anmeldung für publizierte Meldungen verfügbar. Falls der Benutzer als „Kantonaler Administrator“ oder "Kantonale Druckerei“ angemeldet ist, werden auch Meldungen angezeigt, die den Status „Freigegeben“ haben.

Anhang

2.5 Liste der Publikationstypen

Publikations-Typ	Name
OB00	Vorankündigung
OB01	Ausschreibung
OB02	Zuschlag
OB03	Teilnehmerauswahl
OB04	Abbruch
OB05	Ausschreibungen (Zusammenfassung)
OB06	Berichtigung
OB07	Wettbewerb
OB08	Wettbewerbsergebnis
OB09	Widerruf

2.6 Fehlermeldungen

Falls es zu einem Fehler kommt, kann ein Fehlercode und ein Fehlertext ausgelesen werden. Es wird ein generischer „Fault“ zurückgemeldet. Jeder Fault verfügt über die zwei Parameter „Code“ und „Message“.

- Der Fehlercode wird durch die Methode `getFaultInfo.getCode()` zurückgeliefert.
- Die Fehlerbeschreibung wird mit `getFaultInfo.getMessage()` zurückgeliefert.

Folgende Fehlermeldungen gibt es:

Fehlercode	Beschreibung
0	Internal error !
1	You do not have sufficient rights to perform this method !
2	Date out of range !
10	Internal error in getAuthenticationObject!
20	Not authenticated!
21	Internal error in getNoticeList!
30	Error in service getNotice. Not authenticated!

31	Internal Error in getNoticeXml !
32	Error in service getNotice. Notice not available!!
40	Error in service getProviderFromNotice. Not authenticated!
41	Internal error in getProviderFromNotice !
70	Error in service getNoticePdf. Not authenticated !
81	Internal error in getNoticeHtml !
82	Error in service getNoticeHtml. Notice not available!
201	Internal error in getSearchNoticeList !
202	Internal error in getSearchNoticeCount !

2.7 JAVA – Beispiel

Das Beispielprogramm soll den grundlegenden Ablauf einer Programmierung eines Clients für diesen WEB-Service verdeutlichen.

2.8 Generierung von Client Klassen

Für das Programm unten müssen vorab Client Klassen erzeugt werden. Die Klassen werden mit folgendem Kommando erzeugt:

```
wsimport -keep -p [Java Package für die generierten Daten]
-s [Pfad für erzeugten Klassen] https://www.simap.ch/soapserver?wsdl
```

2.8.1 Beispiel - Programm

```
package <your package name>;
import java.net.URL;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.handler.MessageContext;
import <classpath of generated classes (wsimport)>.Fault;
import <classpath of generated classes (wsimport)>.LongArray;
import <classpath of generated classes (wsimport)>.SoapServer;
import <classpath of generated classes (wsimport)>.SoapServerService;

public class Example {
    public Example() {
    }
    public void main(String username, String password) {
        // Verbindung zum Soap Server herstellen
        String servername = "www.simap.ch";
        try {
            URL url = new URL("https://" + servername + "/soapserver?wsdl");
            QName serviceName = new
                QName("http://notice.server.soap.common.exchange.autinform.de/",
                    "SoapServerService");
            SoapServerService service = new SoapServerService(url, serviceName);
            SoapServer servicePort = service.getSoapServerPort();
            /***** Authentication *****/
            Map<String, Object> req_ctx =
                ((BindingProvider)servicePort).getRequestContext();
            Map<String, List<String>> headers = new HashMap<String,
                List<String>>();
            headers.put("username", Collections.singletonList(username));
            headers.put("password", Collections.singletonList(password));
            req_ctx.put(MessageContext.HTTP_REQUEST_HEADERS, headers);

            /*****/
            boolean succesfulAuthentication = servicePort.getAuthentication();
            if (succesfulAuthentication) {
                System.out.println("Authentication successful!");
            }
        }
    }
}
```

```

    } else {
        System.out.println("Authentication not successful!");
        return;
    }
    int pageNo = 1;
    int recordsPerPage = 10;
    String timespanValue = "YEAR";
    String searchXml = "<search pageNo=\"" + pageNo
        + "\" recordsPerPage=\"" + recordsPerPage + "\" >"
        + "<field name=\"" + "TIMESPAN" + "\"><value>" + timespanValue
        + "</value></field></search>";
    System.out.println("getSearchNoticeList(\"" + searchXml + "\")");
    String count = servicePort.getSearchNoticeCount(searchXml);
    System.out.println("count:" + count);
    LongArray arrayList = servicePort.getSearchNoticeList(searchXml);
    for (int i=0; i<arrayList.getItem().size(); i++) {
        long noticeId = arrayList.getItem().get(0);
        System.out.println("id:" + noticeId);
        String xml = servicePort.getNoticeXml(noticeId);
        System.out.println ("xml:" + xml);
        String html = servicePort.getNoticeHtml(noticeId);
        System.out.println ("html:" + html);
    }
} catch (Fault f) {
    f.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

2.9 Beispiel eines SOAP-Requests

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:not="http://notice.server.soap.common.exchange.autinform.de/">
    <soapenv:Header/>
    <soapenv:Body>
        <not:getSearchNoticeCount>
            <searchXml>
                &lt;search pageNo="1" recordsPerPage="20"&gt;
                    &lt;field name="KEYWORD"&gt;&lt;value>Tunnel&lt;/value&gt;&lt;/field&gt;
                    &lt;field name="TYPE_CONTRACT_CD_OB"&gt;&lt;value>WORKS&lt;/value&gt;&lt;/field&gt;
                &lt;/search&gt;
            </searchXml>
        </not:getSearchNoticeCount>
    </soapenv:Body>
</soapenv:Envelope>

```