



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Volkswirtschaftsdepartement EVD
Staatssekretariat für Wirtschaft SECO

Interface SOAP de la FOSC

Version 3.10

Table des matières

1	Introduction	4
2	Description.....	5
2.1	Conditions d'utilisation du service web	5
2.2	Serveur Soap	5
2.3	Appel du service web	5
2.3.1	Fonctions pour une feuille officielle cantonale	5
2.3.1.1	Fonction getActualCantonNoticeList()	6
2.3.1.2	Fonction getActualCantonNoticeListForCategory(category)	6
2.3.1.3	Fonction getNoticeList(publishDate).....	6
2.3.2	Fonctions pour les abonnés	6
2.3.2.1	Fonction getNoticeListForSubscriber(publishDate, subscriptionId).....	7
2.3.2.2	Fonction getNoticeListForSubscriberDateRange(publishDateFrom, publishDateTo, subscriptionId)	7
2.3.3	Fonctions générales	7
2.3.3.1	Fonction getNoticeXml(documentId)	7
2.3.3.2	Fonction getNoticeHtml(documentId)	8
2.3.3.3	Fonction getNoticePdf(documentId)	8
2.3.4	Fonction getNoticeXmIs(publishDate)	8
2.3.5	Fonctions pour la feuille officielle du canton de Zurich	9
2.4	Messages d'erreur	9
2.5	Exemple de programme Java	10

Historique des modifications

Numéro de version	Date	Modification	Pages	Auteur
1.0	15.02.2012	Création	Toutes	J. Quapp
1.1	10.04.2012	Petites corrections et exemple de programme	Toutes	J. Quapp
1.2	24.05.2012	Modifications de détail		J. Quapp
2.0	01.10.2012	Extension du service	Toutes	J. Quapp
2.01	24.10.2012	Corrections du texte		J. Quapp
2.02	20.11.2012	Corrections du texte		J. Quapp
3.0	22.11.2012	Corrections du texte		SECO
3.01	18.08.2013	Nouvelle méthode getNoticeHtml(long docId)		J. Quapp
3.10	10.11.2013	Modification de la méthode getNoticeXML(long docId)		J. Quapp

1 Introduction

La FOSC offre une interface SOAP permettant d'appeler les interfaces de programmation (API). Le présent document contient toutes les informations dont vous avez besoin pour utiliser l'interface SOAP. Si vous n'êtes pas encore enregistré comme utilisateur de la FOSC, vous pouvez le faire sur le site suivant :

<https://www.fosc.ch/SOAP> est un protocole (par http) d'élaboration d'un service web permettant d'accéder aux fonctions offertes par la FOSC depuis un programme tiers rédigé dans un autre langage de programmation.

2 Description

2.1 Conditions d'utilisation du service web

- L'utilisateur du service web doit être inscrit à la FOSC.
- Pour accéder aux fonctions spéciales, l'utilisateur doit disposer de droits supplémentaires.

2.2 Serveur Soap

Pour accéder au serveur Soap, utilisez l'adresse suivante:

<https://www.shab.ch/soapserver>

2.3 Appel du service web

L'authentification doit impérativement s'effectuer dans l'en-tête (*header*) http.

Un message d'erreur est retourné si l'authentification permettant l'accès à la fonction souhaitée n'est pas transmise dans l'en-tête http. Cette authentification requiert deux paramètres : `username` (nom d'utilisateur) et `password` (mot de passe).

L'en-tête http se présente donc comme suit :

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

Le nom d'utilisateur et le mot de passe sont identiques à ceux requis pour l'ouverture d'une session sur la FOSC.

Seule l'authentification permet d'utiliser le service web. Veuillez noter que certains appels de service web exigent des droits spécifiques.

L'authentification peut être vérifiée via la fonction `getAuthentication`.

Fonctions supportées

Toutes les fonctions décrites ci-après sont accessibles dans le cadre du service. Les autres fonctions, à usage interne, ne sont pas utilisables.

2.3.1 Fonctions pour une feuille officielle cantonale

Les feuilles officielles cantonales peuvent charger les annonces récentes dans leur système rédactionnel grâce aux fonctions suivantes. Les annonces peuvent être chargées au format XML.

2.3.1.1 Fonction `getActualCantonNoticeList()`

Description : cette fonction établit la liste de tous les identifiants des annonces validées et publiées, dont la clôture rédactionnelle est intervenue dans les 14 derniers jours.

- Seules les annonces dont le statut est « validé » ou « publié » sont transmises.
- Seules les annonces entrant dans la même rubrique que celle retenue lors de l'inscription de l'imprimerie cantonale sont sélectionnées.

Disponibilité : la fonction n'est proposée que si l'utilisateur a le rôle d' « imprimerie cantonale ».

2.3.1.2 Fonction `getActualCantonNoticeListForCategory(category)`

Arguments :

- `category` code de rubrique à deux caractères. Exemple «KK»=Konkurse (faillites)

Description : cette fonction établit la liste de tous les identifiants des annonces validées et publiées, dont la clôture rédactionnelle est intervenue dans les 14 derniers jours.

- Seules les annonces dont le statut est « validé » ou « publié » sont transmises
- Seules les annonces de la rubrique [`category`] sont sélectionnées. [`category`] peut prendre les valeurs suivantes: « AW », « KK », « NA », « SB », « HR ».

Disponibilité : la fonction n'est proposée que si l'utilisateur a le rôle d' « imprimerie cantonale ».

2.3.1.3 Fonction `getNoticeList(publishDate)`

Arguments :

- `publishDate` date de publication

Description: une imprimerie cantonale reçoit tous les identifiants des documents d'annonces ayant la date de publication indiquée.

- Seules les annonces dont le statut est « validé » ou « publié » sont transmises.
- Seules les annonces appartenant à la même rubrique que celle retenue lors de l'inscription de l'imprimerie cantonale sont sélectionnées.

Disponibilité : la fonction n'est proposée que si l'utilisateur a le rôle d' « imprimerie cantonale ».

2.3.2 Fonctions pour les abonnés

Les abonnés peuvent également recevoir les annonces au format XML via ce service web, pour autant qu'ils aient choisi le format de données correspondant.

2.3.2.1 Fonction `getNoticeListForSubscriber(publishDate, subscriptionId)`

Arguments :

- **publishDate** date de publication
- **subscriptionId** identifiant de votre abonnement. Cette information figure dans la liste de vos abonnements sous Espace clients → Abonnement en ligne - Mes données. Pour les abonnements reconduits, n'importe quel identifiant d'abonnement suffit.

Description : obtention d'une liste des identifiants des annonces. Cette liste est générée à partir de la requête de l'abonné. Les annonces correspondent à celles que l'abonné voit dans son espace client pour la date de publication sélectionnée.

La liste alors générée est celle des identifiants des annonces publiées le jour indiqué et répondant aux critères de recherche saisis par les abonnés.

Disponibilité : la fonction n'est disponible que si l'utilisateur est un « abonné FOSC ».

2.3.2.2 Fonction `getNoticeListForSubscriberDateRange(publishDateFrom, publishDateTo, subscriptionId)`

Arguments :

- **publishDateFrom** date de publication (de...)
- **publishDateTo** date de publication (à...)
- **subscriptionId** identifiant de votre abonnement. Cette information figure dans la liste de vos abonnements sous Espace clients → Abonnement en ligne - Mes données. Pour les abonnements reconduits, n'importe quel identifiant d'abonnement suffit.

Description : obtention d'une liste des identifiants des annonces. Cette liste est générée à partir de la recherche de l'abonné. Les annonces correspondent à celles que l'abonné voit dans son espace client pour la plage de dates de publication sélectionnée [**publishDateFrom**]-[**publishDateTo**].

La liste alors générée est celle des identifiants des annonces publiées le jour indiqué et répondant aux critères de recherche saisis par les abonnés.

Disponibilité : la fonction n'est disponible que si l'utilisateur est un « abonné de la FOSC ».

Veillez à toujours saisir l'identifiant actuel de l'abonnement.

2.3.3 Fonctions générales

Les méthodes ci-après peuvent être utilisées par les différents groupes d'utilisateurs.

2.3.3.1 Fonction `getNoticeXml(documentId)`

Arguments :

- **documentId** numéro d'annonce

Description : cette fonction permet d'obtenir le code XML de l'annonce en entrant l'identifiant de celle-ci.

La description du code XML de l'annonce peut être obtenue en passant par le site www.fosc.ch.

Remarque:

Jusqu'à la version 3.01 le début du code XML se présentait comme suit:

```
<?xml version="1.0" encoding="UTF-8"?>
<FORM ACTIVE.PAGE="1" FOSC_NR="85"
ID="[ID]" LANG="[DE,FR,IT,EN]" PUBLISH_NR="[1,2,3]"
USER="[USER_NUMBER]"> VERSION="[VERSION_NR]"<
<[SUB_CATEGORY]>...
```

A partir de la version 3.1 le code XML se présente de la manière suivante:

```
<?xml version="1.0" encoding="UTF-8"?>
<FORM ACTIVE.PAGE="1"
ID="[ID]" LANG="[DE,FR,IT,EN]" PUBLISH_NR="[1,2,3]"
USER="[USER_NUMBER]" VERSION="[VERSION_NR]">
<[SUB_CATEGORY]>...
```

Disponibilité : la fonction n'est disponible que si l'utilisateur a ouvert une session. L'annonce doit être publiée. Si l'utilisateur est un administrateur cantonal, les annonces ayant le statut « validé » sont également affichées.

2.3.3.2 Fonction getNoticeHtml(documentId)

Arguments :

- **documentId** numéro d'annonce

Description : cette fonction permet d'obtenir le code HTML de l'annonce en entrant l'identifiant de celle-ci, qui est celui qui apparaît lors de la recherche dans la FOSC.

Disponibilité : la fonction n'est disponible que si l'utilisateur a ouvert une session. L'annonce doit être publiée. Si l'utilisateur est un administrateur cantonal, les annonces ayant le statut « validé » sont également affichées.

2.3.3.3 Fonction getNoticePdf(documentId)

Arguments :

- **documentId** numéro d'annonce

Description : on obtient, grâce à cette fonction, le PDF publié de l'annonce avec l'identifiant de celle-ci.

Disponibilité : la fonction n'est disponible que si l'utilisateur a ouvert une session. La FOSC étant la seule à proposer des PDF, cette fonction ne peut être utilisée que sur cette plateforme.

2.3.4 Fonction getNoticeXmIs(publishDate)

Arguments :

- **publishDate** date de publication FOSC

Description : on obtient, grâce à cette fonction, le fichier complet XML. Le fichier complet comprend toutes les annonces du jour de publication indiqué.

Vous devez indiquer comme argument une date de publication FOOSC.

Disponibilité : la fonction n'est disponible que si l'utilisateur a ouvert une session et uniquement pour les éditions publiées.

2.3.5 Fonctions pour la feuille officielle du canton de Zurich

Les fonctions ci-après concernent uniquement la feuille officielle du canton de Zurich et uniquement cette configuration.

- getActualCantonNoticeListForCategoryWithStatus
- getAttachment
- getProviderFromNotice

2.4 Messages d'erreur

En cas d'erreur, un code et un descriptif d'erreur sont générés. Le message générique « Fault » comporte deux paramètres « Code » et « Message ».

Le code d'erreur peut être obtenu via la fonction getFaultInfo.getCode().

La description de l'erreur peut être obtenue via la fonction getFaultInfo.getMessage().

Messages d'erreur possibles :

Code d'erreur	Description
0	Internal error !
1	You do not have sufficient rights to perform this operation !
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!!
35	Internal Error in getNoticeXmIs !
36	Error in service getNoticeXmIs. Not authenticated!
37	Error in service getNoticeXmIs. Date not published!
40	Error in service getProvider. Not authenticated!
41	Internal error in getProviderFromNotice !
42	Internal error in getProvider !

50	Error in service getAttachment. Not authenticated!
51	Internal error in getAttachment !
52	No document in getAttachment !
60	Error in authentication. User is not a subscriber!
61	Error in getNoticeListForSubscription. Subscription id is not valid for the subscriber!
62	Error in getNoticeListForSubscription. Error processing search list!
70	Error in service getNoticePdf. Not authenticated !
71	Internal error in getNoticePdf !
72	No document in getNoticePdf !

2.5 Exemple de programme Java

L'exemple ci-après présente la programmation type d'une application client pour ce service web.

Remarque : des classes de clients doivent être générées avant toute programmation. Pour ce faire, utiliser la commande

```
wsimport -keep -p [Pack Java pour les données générées]
-s [chemin des fichiers sources] https://www.shab.ch/soapserver?wsdl
```

```
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Collections;
import java.util.GregorianCalendar;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;
import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.handler.MessageContext;
import de.autinform.exchange.common.soap.client.notice.Fault;
import de.autinform.exchange.common.soap.client.notice.LongArray;
import de.autinform.exchange.common.soap.client.notice.SoapServer;
import de.autinform.exchange.common.soap.client.notice.SoapServerService;

public class Example {
    private final static SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy",
Locale.getDefault());

    public Example() {
```

```

}

public void main(String username, String password) {
    // Erstellung einer Verbindung zum Soap Server
    String servername = "www.shab.ch";
    int port = 9191;
    try {
        URL url = new URL("http://" + servername + ":" + port + "/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;
        }

        GregorianCalendar c = new GregorianCalendar();
        c.setTime(dateFormat.parse("26.01.2013"));
        XMLGregorianCalendar publishDate =
            DatatypeFactory.newInstance().newXMLGregorianCalendar(c);
        LongArray arrayList = servicePort.getNoticeListForSubscriber(publishDate,
            4711);
        List<Long> noticeList = arrayList.getItem();
        for (int i=0; i<noticeList.size(); i++) {
            long documentId = noticeList.get(i);
            String contentXml = servicePort.getNoticeXml(documentId);
            System.out.println("ID: " + documentId);
            System.out.println("XML: " + contentXml);
        }
    } catch (Fault f) {
        f.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
}  
}
```