

Installation Proxy d'authentification CAS-Zimbra

Pré-requis

Pour son fonctionnement, le Proxy d'authentification CAS-Zimbra nécessite d'avoir au préalable :

- Une installation des binaires Maven en version 2.2.x
- Une installation CAS fonctionnel
- Une installation Zimbra **en version 6.0.x ou 7.x**
- Une installation Tomcat fonctionnel

Téléchargement des sources

Il est possible de récupérer les sources de deux façons distinctes :

- Les [télécharger depuis le site Sourcesup](#) du projet
- Les récupérer directement avec un client Subversion

Téléchargement direct

Chaque version est disponible soit en *.zip* (pour les machines Windows) ou en *.tar.gz* (pour les machines Linux/UNIX).

Une fois l'archive récupérée, il faut la décompresser sur la machine où l'on souhaite installer la portlet :

```
tar zxvf web-auth-zcs-1.2.X-src.tar.gz
```

Client Subversion

Le [dépôt Subversion](#) est ouvert en anonyme, il est recommandé de récupérer une version présente dans la branche 'tags' :

```
svn co https://subversion.cru.fr/web-auth-zcs/tags/1.2.X/ web-auth-zcs-1.2.X
```

Configuration

La configuration de l'application doit être réalisée dans un fichier *profiles.xml* présent à la racine des sources. Un fichier d'exemple *profiles-sample.xml* est fourni pouvant servir de base à toute nouvelle installation :`cp profiles-sample.xml profiles.xml`

Voici les propriétés devant être renseignées dans ce fichier :

- **deploy.dir** : répertoire où sera déployée l'application. Il s'agit d'un dossier qui se trouvera dans le répertoire *webapps*. Il est recommandé de conserver à la fin de ce dossier le numéro de version de l'application et d'utiliser ensuite un lien symbolique pour préciser la version active, afin de faciliter les futures mises à jour.
- **public.host** : nom d'hôte du serveur hébergeant l'application
- **public.port** : port http hébergeant l'application (vide si standard 80 ou 443)
- **cas.server** : il s'agit de l'URL du serveur CAS, par exemple <http://cas.univ.fr>, ou encore <http://cas.univ.fr/cas>.
- **zcs.zimbraAdminUser** : nom d'un utilisateur zimbra pouvant demandé des token pour un autre utilisateur zimbra
- **zcs.zimbraAdminPass** : mot de passe de cette utilisateur
- **zcs.zimbraAdminUrl** : url d'administration zimbra pour faire des request SOAP
- **zcs.tokenLifeTime** : durée de vie du token en seconde
- **log4j.level.main** : cette propriété doit contenir la valeur *error* pour une installation de production.
- **log4j.level.spring** : cette propriété doit contenir la valeur *warning* pour une installation de production.
- **log4j.level.app** : cette propriété doit contenir la valeur *info* pour une installation de production.
- **zcs.domain (DEPRECATED en version >= 1.2)** : domaine zimbra utilisé pour compléter le nom de utilisateur après l'authentification CAS.

Les logs de l'application (ne contenant aucune information importante, si ce n'est les exceptions pouvant se produire) peuvent être configurés de trois façons différentes, simplement en commentant/décommentant les sections correspondantes du fichiers *profiles.xml*.

Logs console

Dans ce mode, les logs sont écrits directement sur la sortie standard et aboutissent généralement dans le fichier *catalina.out* de Tomcat. Aucune configuration n'est nécessaire avec ce mode.

Logs syslog

Propriétés à renseigner :

- **log4j.syslog.host** : cette propriété doit contenir le nom du serveur syslog.
- **log4j.syslog.facility** : cette propriété doit contenir la *facility* syslog à utiliser.

Logs fichier

Propriétés à renseigner :

- **log4j.file.file** : cette propriété doit contenir le chemin du fichier de log.
- **log4j.file.nb** : cette propriété doit contenir le nombre de fichiers d'archives qui seront conservés
- **log4j.file.size** : cette propriété doit contenir la taille maximale d'un fichier de log, provoquant sa rotation vers un fichier d'archive.

Les logs de l'application (ne contenant aucune information importante, si ce n'est les exceptions pouvant se produire) peuvent être configurés de trois façons différentes, simplement en commentant/décommentant les sections correspondantes du fichiers *profiles.xml*.

Déploiement

```
cd web-auth-zcs-1.2.X
mvn package
```

configuration tomcat

```
<Context path="/auth-zcs" docBase="/home/webpass/web-auth-zcs" reloadable="false">
  <Manager pathname="" />
</Context>
```

Utilisation

Si vous proxy auth zimbra est installé sur <http://auth-zcs.univ.fr> vous pouvez accéder à:

```
http://auth-zcs.univ.fr/univ.fr.token => pour obtenir un token dans le domain univ.fr

http://auth-zcs.univ.fr/etudiant.univ.fr.token => pour obtenir un token dans le domain etudiant.univ.fr
...
```