

1- Installation

Pré-requis

Pour son fonctionnement, la portlet Zimbra nécessite d'avoir au préalable :

- Une installation des binaires Maven en version 2.2.x ou 3.0.x
- Un portail fonctionnel **en version 3.0.x, 3.1.x, 3.2.x ou 4.0.x**
- Une installation Zimbra **en version 7.0.x/8.x (testé en 8.6)**
- Une installation du connecteur d'authentification CAS/Zimbra

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 nancy2-portlet-zimbra-3.4.3-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.renater.fr/portlet-zimbra/tags/3.4.3 portlet-zimbra-3.4.3
```

Configuration

La configuration de la portlet doit être réalisée dans un fichier *portlet-zimbra.properties* présent à la racine des sources. Un fichier d'exemple est fourni pouvant servir de base à toute nouvelle installation.

Voici les propriétés devant être renseignées dans ce fichier :

- **deploy.dir** : répertoire où sera déployée la portlet. Il s'agit d'un dossier qui se trouvera certainement à côté de celui de l'ENT, dans le répertoire *w ebapps*. Il est recommandé de conserver à la fin de ce dossier le numéro de version de la portlet et d'utiliser ensuite un lien symbolique pour préciser la version active, afin de faciliter les futures mises à jour.
- **portlet.title/portlet.title.short** : c'est le nom qui apparaîtra dans le portail aux utilisateurs une fois la portlet déployée et publiée.
- **public.host** : cette propriété doit être renseignée à la **même valeur que celle du portail**. En effet la portlet étant proxy CAS, cette information est nécessaire au calcul de l'URL de service.
- **real.host** : cette propriété doit être renseignée à la **même valeur que celle du portail**. En effet la portlet étant proxy CAS, cette information est nécessaire au calcul de l'URL de dépôt du PGT. Si vous travaillez dans un environnement mono-portail, elle doit avoir la même valeur que la propriété précédente. Si vous travaillez dans un environnement multi-serveur avec répartition de charge, elle doit être spécifique à l'instance de chaque serveur du cluster.
- **deploy.uri** : il est conseillé de laisser cette propriété à sa valeur par défaut. Voir la définition du contexte Tomcat.
- **cas.server** : il s'agit de l'URL du serveur CAS, par exemple <http://cas.univ.fr>, ou encore <http://cas.univ.fr/cas>.
- **cas.callback.uri** : cette propriété ne devrait pas être modifiée, la valeur par défaut convient à toutes les installations.
- **zimbra.token.service.url** : cette propriété doit contenir l'URL du connecteur d'authentification CAS/Zimbra.
- **zimbra.soap.service.url** : cette propriété doit contenir l'URL du service SOAP Zimbra, habituellement accédé en HTTPs à l'URI */service/soap* de votre serveur Zimbra.
- **zimbra.url** : cette propriété doit contenir l'URL de votre serveur Zimbra, accédée par vos utilisateurs pour consulter leur messagerie. Elle est utilisée pour proposer des liens directs vers Zimbra depuis la portlet.
- **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.main** : cette propriété doit contenir la valeur *error* pour une installation de production.

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 fichier *portlet-zimbra.properties*.

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.

Déploiement

La compilation et le déploiement se font à l'aide de la commande suivante :

```
mvn package
```

Si tout s'est bien passé (**build successful**), vous devriez voir les fichiers de l'application dans le dossier choisi pour le déploiement (**deploy.dir**). Si vous avez choisi de conserver le numéro de version dans le nom du répertoire de déploiement, il est recommandé de faire un lien symbolique vers la version courante (cela simplifiera considérablement les futures mises à jour).

Attention, si vous avez un portail en 4.0x, commenter les lignes suivantes dans le fichier **src/main/webapp/WEB-INF/web.xml** :

```
<servlet-mapping>
    <servlet-name>zimbra</servlet-name>
    <url-pattern>/PlutoInvoker/zimbra</url-pattern>
</servlet-mapping>
```

Un petit lien symbolique :

```
ln -s portlet-zimbra-3.4.3 portlet-zimbra
```

La dernière étape consiste à définir ce nouveau contexte dans la configuration de Tomcat. Il faut ajouter les informations suivantes au fichier *server.xml* en prenant soin de remplacer les propriétés **deploy.dir** et **deploy.uri** par les valeurs choisies pour la portlet (ou par le chemin du lien symbolique si vous en avez défini un) :

```
<Context path="deploy.uri" docBase="deploy.dir" crossContext="true">
  <Manager pathname="" />
</Context>
```

Important : cette portlet utilise pour la partie Ajax le partage de session entre portlet et servlet. Pour que celui-ci fonctionne, vous devez absolument définir la propriété *emptySessionPath* sur le(s) connecteur(s) que vous utilisez dans Tomcat. Voici un exemple sur le connecteur JK, applicable également au connecteur HTTP :

```
<Connector port="8009" protocol="AJP/1.3"
    redirectPort="4443" emptySessionPath="true" />
```

Publication

Voici un fichier de publication d'exemple pour cette portlet, il est nécessaire de remplacer les propriétés correspondant à celles saisies dans le fichier *portlet-zimbra.properties* :

```

<channel-definition script="classpath://org/jasig/portal/io/import-channel_v2-6.crn">
<title>Zimbra</title>
<name>Zimbra</name>
<fname>portlet-zimbra</fname>
<desc>Zimbra</desc>
<type>Portlet</type>
<class>org.jasig.portal.channels.portlet.CSpringPortletAdaptor</class>
<timeout>30000</timeout>
<hasedit>N</hasedit>
<hashelp>N</hashelp>
<hasabout>N</hasabout>
<secure>N</secure>
<locale>en_US</locale>
<categories>
  <category>CATEGORIE</category>
</categories>
<groups>
  <group>GROUPE</group>
</groups>
<parameters>
  <parameter>
    <name>portletApplicationId</name>
    <value>deploy.uri</value>
    <description/>
    <ovrd>N</ovrd>
  </parameter>
  <parameter>
    <name>portletName</name>
    <value>zimbra</value>
    <description/>
    <ovrd>N</ovrd>
  </parameter>
</parameters>
</channel-definition>

```

Voici les trois paramètres à spécifier dans ce fichier :

- **CATEGORIE** : la catégorie de canaux dans laquelle doit apparaître la portlet
- **GROUPE** : le ou les groupe(s) qui auront accès à la portlet
- **deploy.uri** : même valeur que dans le fichier *portlet-zimbra.properties*

Il ne reste plus qu'à publier la portlet dans le portail.

svn co <https://subversion.cru.fr/portlet-zimbra/tags/3.4.3> portlet-zimbra-3.4.3