

Installation Gestion d'attribut LDAP

Canal PréférencesVersion 2

Installation, paramétrage du canal du canal
Auteur : Sébastien GAUDIN ([Université nancy 2](#))

- [Canal PréférencesVersion 2](#)
- [Installation du MAG](#)
- [Configuration du canal](#)
 - [Le fichier de configuration](#)
 - [Restreindre l'accès à des groupes uPortal](#)
 - [Les actions MAG](#)
 - [La partie LDAP](#)
 - [Options diverses](#)
 - [Les formats](#)
 - [boolean check \(String before\)](#)
 - [String transform \(String before\)](#)
 - [display \(String\[\] before\)](#)
 - [String\[\] exemple \(\)](#)
- [Les fichiers xsl](#)
 - [Les imports](#)
 - [L'affichage d'un attribut](#)
 - [La saisie d'un attribut](#)
- [Déploiement](#)

Installation du MAG

Ce canal est dépendant du MAG (testé en versions 1 et 2).

Configuration du canal

Le fichier de configuration

Voir la documentation du fichier de [configuration du MAG](#), également sur son plugin [RestrictedAction](#) car certaines action MAG sont filtrées par groupe.

Restreindre l'accès à des groupes uPortal

Il est possible de bloquer certaines action à un (ou plusieurs) groupe uPortal. Pour cela, il faut commencer par les déclarer dans le fichier de configuration.

```
<groupmapping>
  <group localname="PERS"
        uportalid="local.106"/>
  <group localname="ETUD"
        uportalid="local.103"/>
</groupmapping>
```

Par exemple, les seuls groupes que nous utiliserons sont "local.106" que nous appellerons "PERS" dans le reste du fichier de configuration et "local.103" que nous appellerons "ETUD".

Les actions MAG

Comme le canal se base sur le MAG, il faut définir les actions.

Il faut en premier lieu définir l'action par défaut.

```

<!-- ACTION CENTRALE -->
    <action name="default"
        classname="org.esupportail.portal.channels.CPref.Menu"
        rendertype="xsl">
        <param name="PERS">
            <value>menu_personnel.xsl</value>
        </param>
        <param name="ETUD">
            <value>menu_etudiant.xsl</value>
        </param>
    </action>

```

Cette action pointe sur la page la classe gérant le menu. Vous pouvez, comme le montre l'exemple, avoir une page xsl de menu par groupe que vous avez déclaré ci-dessus.

Pour les autres actions, il faut positionner les attributs LDAP qui seront utilisés :

```

<!-- Information personnel pour un membre du personnel -->
<action name="infoPers"
    classname="org.esupportail.portal.channels.CPref.actions.Default"
    sslfile="infoPers.ssl" rendertype="ssl"
    log="info">
    <param name="plug.esup.restrictedAction">
        <value>PERS</value>
    </param>
    <param name="attributeRead">
        <value>uid</value>
        <value>givenname</value>
        <value>sn</value>
        <value>supannorganisme</value>
        <value>postaladdress</value>
    </param>
    <param name="attributeWrite">
        <value>telephonenumber</value>
        <value>facsimiletelephonenumber</value>
        <value>description</value>
        <value>supanncivilite</value>
        <value>labeleduri</value>
    </param>
    <param name="attributeXsl">
        <value>uid</value>
    </param>
</action>

```

Dans cette action, nous pouvons voir que l'accès est restreint aux membres de "PERS". Cette restriction est bien entendue optionnelle.

La liste des attributeRead définis les attributs qui seront lus dans le LDAP pour l'affichage. La liste des attributeWrite définis les attributs qui seront lus et qui seront modifiable par l'utilisateur. La liste des attributeXsl définis les attributs qui seront envoyés à la feuille xsl (pour faire des testes sur la valeur par exemple).

La partie LDAP

Dans cette partie, vous allez préciser les paramètres LDAP que vous allez utiliser. Entrez tout d'abord les informations du serveur, dans le tag "ldap", en séparant les adresses accessibles pour la lecture et celle accessible pour l'écriture.

```

<ldap urlRead="ldap://ldap.univ.fr:392 ldap://ldap2.univ.fr:392" urlWrite="ldap://ldap.univ.fr:392"

```

Ensuite, précisez la base de recherche des personnels.

```
baseDn="ou=People,dc=univ,dc=fr"
```

La dernière chose à préciser dans les informations du serveur est le filtre de recherche d'une personne. Indiquez par un %u l'endroit où le login devra être placé.

```
filtreDn="(&(objectclass=inetOrgPerson)(uid={u}))" >
```

Vous pouvez préciser aussi les bindDn passDn pour la connexion "anonyme". Pour cela, saisissez le bindDn (dn complet) et le passDn.

A l'intérieur de ce tag ldap, décrivez la liste des attributs que vous voulez formater. Pour affecter un format à un attribut, précisez le nom ldap de l'attribut et la classe correspondante :

```
<attribut ldapName="facsimiletelephonenumber" format="org.esupportail.portal.channels.CPref.formats.supann.
TelephoneNumber" />
```

Soit vous avez besoin de différencier le format selon les différents groupes uPortal déclarés plus haut. Dans ce cas, vous ne devez pas remplir l'attribut "format" comme ci-dessus, mais créer un sous tag "format" par groupe.

```
<attribut ldapName="n2atrmalldrop" >
    <format group="PERS" name="org.esupportail.portal.channels.CPref.formats.nancy2.RedirectionPersonnelN2"
/>
    <format group="ETUD" name="org.esupportail.portal.channels.CPref.formats.nancy2.RedirectionEtudiantN2" />
</attribut>
```

Options diverses

```
<divers passwordInSession="no" />
```

Avec cette option, l'utilisateur devra saisir son mot de passe avant chaque panneau de modification. Par contre, avec la valeur "yes", l'utilisateur saisie une seule fois son mot de passe pour la durée de la session uportal.

Les formats

Des formats standards sont fournis avec le package. Ils se trouvent dans le package org.esupportail.portal.channels.CPref.formats.supann. Vous pouvez en écrire d'autres. Je conseille de les mettre dans un package séparé, pour les repérer plus facilement : exemple org.esupportail.portal.channels.CPref.formats.univ.nomFormat

Les formats que vous définissez doivent hériter de la classe org.esupportail.portal.channels.CPref.formats.Format. Cette classe propose différentes méthodes abstraites que vous devrez re définir dans vos formats. Chacune des méthodes décrites ci-dessous (sauf "exemple") peut avoir en paramètre un tableau de valeur. Ce cas de figure est pris en charge directement par la classe Format. Cela signifie que vous n'avez qu'à spécifier les caractéristiques pour une valeur et lors de la vérification de plusieurs valeurs, les appels sont faits automatiquement.

boolean check (String before)

Cette méthode décrit les différentes expressions régulières acceptées lors de la saisie et retourne true si le format est correct, false sinon.

String transform (String before)

L'utilisateur peut avoir la possibilité de saisir la valeur dans différents formats. Or il faut normaliser les entrées dans le LDAP. Pour cela, avant chaque écriture d'attribut, cette méthode est appelée et transforme la saisie utilisateur dans un format spécifique.

display (String[] before)

Lors de l'affichage d'une valeur d'un attribut, il y a un appel à cette méthode qui transforme la valeur enregistrée dans LDAP en une valeur plus conviviale pour l'utilisateur.

String[] exemple ()

Vous pouvez décrire ici autant d'exemple que vous voulez. Ils apparaîtront à coté du champs lors de la saisie.

Les fichiers xsl

Les imports

L'import de "common" permet d'avoir accès aux templates d'affichage des attributs. Dans la feuille errorBar, sont définis les messages d'erreur à afficher si la saisie se passe mal.

```
<xsl:import href="../../errorBar.xsl" />
<xsl:import href="../../common.xsl" />
```

L'affichage d'un attribut

Dans une ligne d'un tableau html, appliquez le template comme dans l'exemple ci-dessous :

```
<tr>
  <xsl:apply-templates select="INFOS/ATTRIBUTE[@name='uid']">
    <xsl:with-param name="label">Login</xsl:with-param>
  </xsl:apply-templates>
</tr>
```

Appliquez le template ATTRIBUTE mais en ne sélectionnant que l'attribut dont le nom est "uid". Ceci permet de positionner le label lors de l'affichage. L'intérêt de passer par ce template est que les styles css seront mis automatiquement.

La saisie d'un attribut

Vous devez appliquer le même template que pour l'affichage mais en précisant des paramètres supplémentaires.

```
<tr>
  <xsl:apply-templates select="INFOS/ATTRIBUTE[@name='telephonenumber']">
    <xsl:with-param name="label">Téléphone</xsl:with-param>
    <xsl:with-param name="modifiable">true</xsl:with-param>
    <xsl:with-param name="saisie">true</xsl:with-param>
    <xsl:with-param name="allowMultiValue">true</xsl:with-param>
  </xsl:apply-templates>
</tr>
```

Déploiement

Préparation du fichier de configuration ConfigChannel.xml.: mettre les renseignements relatifs aux attributs qu'il faut gérer

Préparation du déploiement : modifier les chemins d'installation du fichier build.properties.

Si vous aviez déjà installé une ancienne version du canal, lancez la commande **ant undeploy** , pour effacer toute trace du canal dans l'environnement de production.

Lancement de la commande **ant deploy**, pour déployer le canal dans les bons répertoires.

L'administrateur peut maintenant publier les canaux dont il a besoin en fonction du paramètre d'instanciation : "serverView". Pour cela, il y a 2 méthodes :

La plus simple étant de publier le canal de manière normale, par l'interface WEB prévue à cet effet.

La seconde étant d'utiliser la commande **ant pubchan**.

Cette méthode remplace toute la démarche de publication.

Utilisez la commande ant deploypubchan qui va déposer le fichier de déploiement (pubchan_CPref.xml) dans le répertoire {uPortal}

/properties/chanpub. Les informations nécessaires à la publication sont décrites dans des tags XML, vous permettant ainsi même de spécifier l'attribut "serverView".

Ensuite, lancez la commande "ant pubchan -Dchannel=pubchan_CPref.xml". depuis le répertoire "racine esup".

Vous pouvez bien sur créer autant de fichier xml que vous désirez. Par exemple, rien n'interdit d'en avoir un pour la vue en anonyme, un pour les étudiants et un pour le personnel.