

uPortal – Administration

Objectifs :

- Installer un portail fonctionnant de façon autonome puis le brancher progressivement aux différents composants du système d'information : annuaire LDAP, serveur d'authentification CAS, frontal Apache.
- Se familiariser avec uPortal : gérer les préférences d'affichage, personnaliser l'environnement par défaut, gérer les groupes, les canaux et les fragments.

I. Installation du JDK :

Installer le Java Development Kit dans le répertoire D:\JDK

II. Installation de la distribution quick-start d'uPortal :

Décompresser l'archive `uPortal_rel-2-4-3-quick-start.zip` dans le répertoire de travail D:\. Un dossier `uPortal_rel-2-4-3-quick-start` est automatiquement créé.

Récupérer les fichiers `ant.bat`, `env.cmd`, `start.cmd`, `stop.cmd` et les placer à la racine du dossier uPortal.

Editer le fichier `ant.bat` :

Modifier la variable `JAVA_HOME` pour qu'elle pointe vers D:\JDK

Modifier la variable `INSTALL` pour qu'elle pointe vers D:\uPortal_rel-2-4-3-quick-start

Editer le fichier `env.cmd` :

Modifier la variable `JAVA_HOME` pour qu'elle pointe vers D:\JDK

Modifier la variable `CATALINA_HOME` pour qu'elle pointe vers D:\uPortal_rel-2-4-3-quick-start\Tomcat_5-0-28

III. Configuration de la base de données

Copier le fichier `mysql-connector-java-3.0.15-ga-bin.jar` dans le répertoire `lib` de la distribution uPortal ainsi que dans le répertoire `common/lib` de Tomcat.

Adapter le fichier `build.properties` afin de déclarer cette nouvelle bibliothèque comme pilote JDBC.

Editer le fichier `properties/rdbm.properties` et configurer une nouvelle entrée pour MySQL :

```
driver : com.mysql.jdbc.Driver
url : jdbc:mysql://[HOST]/portalXX
identifiant : portaldev
password : portalN2
```

Editer le fichier `properties/uPortal.xml` afin de configurer les pools de connexions Tomcat PortalDb et PersonDb afin qu'ils utilisent la même base de données.

Editer le fichier `properties/PersonDirs.xml` afin de configurer la récupération d'attributs afin d'utiliser la même base de données.

Editer le fichier `properties/portal.properties` et modifier la valeur de la propriété `getDatasourceFromJndi` afin de ne pas utiliser les connexions Tomcat (nécessaire pendant toute la partie configuration / initialisation de la base).

Exécuter la tâche ANT 'dbtest' afin de vérifier que la base de données a été correctement configurée et noter la valeur de la propriété 'database-version'.

Editer le fichier `properties/db/dbloader.xml` et dupliquer une entrée MySQL, remplacer la valeur 'db-version' par celle notée précédemment.

Exécuter la tâche ANT 'initportal' afin de réaliser plusieurs opérations :

- déploiement dans Tomcat
- initialisation de la base de données
- publication des canaux
- création des environnements pour les utilisateurs (fragments)
- déploiement des portlets

IV. Démarrage

Editer le fichier `properties/Logger.properties` afin de configurer l'emplacement du fichier de log.

Démarrer le portail en lançant la commande `start.cmd`.

Tester le portail avec un navigateur en utilisant l'URL <http://localhost:8080/uPortal> (attention à la casse) et le compte local admin / admin.

Remarques :

- *En cas de problème, toujours penser à regarder le fichier `portal.log`.*
- *A chaque modification d'un fichier de configuration il est nécessaire de relancer la tâche ANT 'deploy'.*
- *Dans la configuration actuelle, seuls les utilisateurs locaux au portail peuvent se connecter. Tester l'ajout d'un nouvel utilisateur à l'aide de la commande*

```
ant md5passwd -Dusername=newuser
```

ainsi que la modification du mot de passe administrateur avec la commande

```
ant md5passwd -Dusername=admin
```

V. Interface et navigation :

Démarrer le portail en lançant la commande `start.cmd`.

Se loguer sur le portail avec un navigateur en utilisant l'URL <http://localhost:8080/uPortal> en tant qu'administrateur `admin / admin` (ou autre si vous avez changé le mot de passe administrateur).

Identifier les éléments : Onglets, Colonnes et Canaux.

Tester les options : retour à la page d'accueil, plan du site (les autres options seront vues plus en détail par la suite).

Enfin, observer les rôles des boutons suivants :



VI. Préférences utilisateur :

Accéder aux préférences par le bouton 

Ajouter un onglet que l'on nommera « essai » et y ajouter deux colonnes.

Souscription d'un canal :

On va maintenant ajouter un canal dans l'une d'entre elles :

Cliquer sur « ajouter un canal »




Souscrire le canal « Person Attributes » disponible dans la catégorie « uPortal ».

Enfin choisissez l'emplacement du canal.

Déplacer maintenant le canal dans l'autre colonne.

Supprimer la colonne vide.

Remarques :

- Le bouton  indique les endroits où peuvent être positionnés les éléments.
- Remarque : Penser à enregistrer les préférences () avant de quitter () sinon elles seront perdues.

Choix de la skin :

Tester le changement de skin du portail en cliquant sur « Skins »

Choix de la langue :

Cliquez sur « langues » et choisir allemand, suédois ou japonais on constate que le contenu du canal « Person Attributes » a changé.

Remarques :

- *Lorsqu'on remet la langue à « français » le texte apparaît en anglais. En effet, il n'y a pas de XSL correspondant à la langue française pour ce canal c'est donc la XSL par défaut qui est choisie.*
- *La personnalisation des fragments sera vue plus tard.*

VII. Branchement de l'authentification LDAP :

Editer le fichier `properties/ldap.xml`.

Créer une nouvelle connexion en utilisant les paramètres suivants :

```
name : Test
host : [HOST]
port : 10389
baseDN : ou=People,dc=univ-nancy2,dc=fr
managerDN : (aucun, bind anonyme)
managerPW : (aucun, bind anonyme)
uidAttribute : uid
protocol : (aucun, pas de SSL)
```

Editer le fichier `properties/security.properties`.

Définir une chaîne d'authentification afin de réaliser les opérations suivantes :

```
root : UnionSecurityContextFactory
root.ldap : SimpleLdapSecurityContextFactory
root.db : SimpleSecurityContextFactory
```

Ne pas oublier de renseigner la propriété indiquant quelle entrée du fichier `ldap.xml` utiliser (Test dans notre cas).

Exécuter la tâche ANT 'deploy' pour prendre en compte les modifications, redémarrer le portail et tester la connexion avec un compte de l'annuaire LDAP (ens1 / ens1).

Remarques :

- *Avant la connexion, le portail ne connaissait pas l'utilisateur ensXX. Ce compte a été automatiquement créé car l'authentification a réussi et le portail fonctionne en mode autocréation.*
- *Ce nouveau compte hérite de toutes les propriétés et de tous les droits du compte 'demo'. Dans le cas particulier de la distribution uPortal, le compte demo a des droits assez importants.*

VIII. Branchement de la récupération d'attributs LDAP :

Editer le fichier `properties/PersonDirs.xml`.

Ajouter une source de données LDAP en utilisant l'un des modèles proposés. Il suffit de faire référence à l'annuaire paramétré dans le fichier `properties/ldap.xml`.

Pour les mappings d'attributs, utiliser ceux fournis dans le fichier `PersonDirs.txt`.

Tester que lors de la connexion avec un utilisateur de l'annuaire, vous avez désormais affiché son nom et son prénom, on peut également vérifier les informations récupérées dans le canal 'Person Attributes'.

IX. Utilisation d'un frontal Apache :

Pour pouvoir utiliser l'authentification CAS en mode proxy, il est impératif que le portail soit accessible en HTTPS. Il est beaucoup plus simple de laisser ce travail à un frontal Apache plutôt que de configurer Tomcat pour le faire.

Editer le fichier `properties/uPortal.xml` et modifier le contexte `uPortal` en `portalXX`. Vérifier dans le fichier de configuration principal de Tomcat (`conf/server.xml`) que le connecteur AJP est bien déclaré sur le port 8009.

Tester l'accès à votre portail en passant par l'URL [http://\[HOST\]/portalXX](http://[HOST]/portalXX) puis en HTTPS.

X. Branchement de l'authentification CAS :

Copier les librairies `casclient.jar` et `cas-security-provider.jar` dans le répertoire `lib`.

Modifier le fichier `build.properties` pour prendre en compte ces deux librairies supplémentaires.

Adapter la target 'deploy' du fichier `build.xml` pour que ces librairies soient correctement déployées.

Editer le fichier `webpages/WEB-INF/web.xml` pour ajouter les informations du fichier `cas.txt`. Attention, ces informations doivent être placées au bon endroit.

Editer le fichier `properties/security.properties` afin d'ajouter l'authentification CAS :

```
root.cas=edu.yale.its.tp.portal.security.YaleCasFilteredContextFactory
```

Ajouter le ticket CAS qui remplace le mot de passé habituel :

```
credentialToken.root.cas=ticket
```

Ajouter l'URL de redirection CAS lors du logout :

```
logoutRedirect.root.cas=https://[HOST]/cas/logout?service=http://[HOST]/portalXX
```

Il faut maintenant indiquer au portail qu'il doit faire confiance au certificat du serveur CAS (ici c'est un certificat auto-signé, dans une architecture de production, il faut fournir le certificat de l'autorité racine au format Java).

Télécharger le fichier `trust.keystore` à l'URL suivante : [http://\[HOST\]/cert/](http://[HOST]/cert/)

Le déposer à la racine du package, avec les fichiers `start.cmd` et `env.cmd`.

Adapter le fichier `env.cmd` pour que la variable `TRUST_CERT` pointe sur ce fichier.

Adapter le fichier `start.cmd` pour préciser au démarrage de Tomcat la confiance dans ce certificat.

Il nous reste à adapter légèrement le canal de login pour qu'il affiche un lien vers le serveur CAS.

Editer le fichier :

`webpages/stylesheets/org/jasig/portal/channels/CLogin/html.xsl`.

Ajouter après la ligne 65 (bouton submit) les deux lignes suivantes :

```

<a
href="https://[HOST]/cas/index.jsp?service=http://[HOST]/porta
lXX/Login">CAS NetID</a>
```

XI. Les Groupes :

Nous allons maintenant voir les 3 façons de créer des groupes uPortal.

Les groupes locaux :

Cette gestion n'est possible qu'à travers le portail

Se connecter au portail en administrateur et accéder au group manager.

Ajouter un groupe en cliquant sur  du groupe parent ou du root « Tous les groupes de personnes ».

Y ajouter un utilisateur en faisant une recherche : admin par exemple.

Penser à prendre en compte les modifications en cliquant sur .

Remarque :

Une fois créé il n'est pas possible de déplacer le groupe, on peut uniquement le supprimer.

Les groupes LDAP :

Il est fortement déconseillé d'utiliser cette création de groupe, nous allons cependant voir comment procéder sur un petit groupe.

Le mapping :

Nous allons d'abord devoir réaliser le mapping entre les attributs LDAP et les attributs uPortal.

Dans :

`properties\PersonDirs.xml`

Adapter les paramètres de connexion LDAP :

```
<url>ldap://[HOST]:10389/ou=People,dc=univ-nancy2,dc=fr</url>
<logonid></logonid>
```

```
<logonpassword></logonpassword>
<uidquery>(uid={0})</uidquery>
<usercontext></usercontext>
```

Faire correspondre les éléments suivants (LDAP)

```
<attribute>
  <name>n2atrtypepeople</name>
  <alias>typepeople</alias>
</attribute>
```

Remarques :

- Le « name » est l'attribut source de données et l' « alias » est l'attribut uPortal
- Attention à la casse.

LDAPGroupStoreConfig.xml

Dans :

properties\groups\LDAPGroupStoreConfig.xml

Créer un groupe de ce type contenant un simple sous-groupe.

```
<LDAPGroupStore>
  <config>
    <url>ldap://[HOST]:10389/ou=People,dc=univ-
nancy2,dc=fr</url>
    <logonid></logonid>
    <logonpassword></logonpassword>
    <keyfield>uid</keyfield>
    <namefield>cn</namefield>
    <usercontext>ou=people</usercontext>
    <refresh-minutes>120</refresh-minutes>
  </config>
  <group name="LDAP Groups" key = "all">
    <description>groupe issu ldap pas bien</description>
    <group name="Personnel LDAP Group" key="1">
      <description>les personnels</description>
      <entity-set>
        <filter
string="n2atrtypepeople=PERS_EMP"/>
      </entity-set>
    </group>
  </group>
</LDAPGroupStore>
```

Utiliser ce filtre qui contient un nombre d'utilisateurs restreint.

Puis dans :

properties\groups\compositeGroupServices.xml

Décommenter alors la partie qui correspond au service LDAP :

```
<service><name>ldap</name> ...
```

Exécuter la tâche ANT 'deploy' pour prendre en compte les modifications, relancer le portail.

Le groupe nouvellement créé n'est pas rattaché à l'arborescence des groupes. Il vous faut alors l'ajouter comme un membre (c'est-à-dire avec une recherche) grâce au group manager du portail.

Il apparaît maintenant dans votre arborescence avec les utilisateurs qu'il contient. Vous constaterez peut être que l'affichage est long et comprendrez pourquoi cette méthode n'est pas envisageable pour des groupes contenant beaucoup d'utilisateurs.

Les groupes PAGES :

L'appartenance à un groupe est calculée, à la connexion de l'utilisateur, suivant ses attributs uPortal de personne. On utilisera pour cela le mapping (vu précédemment). Par conséquent, ce service est incapable de lister les membres d'un groupe.

PAGSGroupStoreConfig.xml

Dans :

\properties\groups\PAGSGroupStoreConfig.xml

Créer deux groupes avec un XML ce type :

```
<Group-Store>
  <group>
    <group-key>TousPers</group-key>
    <group-name>LDAP Tout le personnel enseignant</group-name>
    <group-description>Tout le personnel enseignant de
l'etablissement issu de LDAP</group-description>
    <selection-test>
      <test-group>
        <test>
          <attribute-name>typepeople</attribute-name>
          <tester-
class>org.jasig.portal.groups.pags.testers.StringEqualsIgnoreC
aseTester</tester-class>
          <test-value>PERS_ENS</test-value>
        </test>
      </test-group>
    </selection-test>
  </group>

  <group>
    <group-key>TousEtud</group-key>
    <group-name>LDAP Tous les etudiants</group-name>
    <group-description>Tous les etudiants de l'etablissement
issu de LDAP</group-description>
    <selection-test>
      <test-group>
        <test>
```



```

        <attribute-name>typepeople</attribute-name>
        <tester-
class>org.jasig.portal.groups.pags.testers.StringEqualsIgnoreC
aseTester</tester-class>
        <test-value>APO_N2</test-value>
        </test>
    </test-group>
</selection-test>
</group>
</Group-Store>

```

compositeGroupServices.xml

Ouvrir le fichier :

```
\properties\groups\compositeGroupServices.xml
```

Décommenter alors la partie qui correspond au service PAGES :

```
<service><name>pags</name> ...
```

Exécuter la tâche ANT 'deploy' pour prendre en compte les modifications, relancer le portail.

Le groupe nouvellement créé n'est pas rattaché à l'arborescence des groupes. Il vous faut alors l'ajouter comme un membre (c'est-à-dire avec une recherche) grâce au group manager du portail.

Il apparaît maintenant dans votre arborescence en revanche les utilisateurs qu'il contient ne sont pas affichés. En revanche, si vous cherchez un utilisateur censé appartenir au groupe et que vous affichez ses propriétés vous verrez le groupe apparaître. En effet, c'est au moment de la connexion de l'utilisateur ou lorsqu'il va chercher à accéder à une ressource protégée que le portail vérifiera son appartenance ou non au(x) groupe(s) autorisé(s).

Remarque :

Pour les groupes PAGES comme pour les groupes LDAP, il est possible de faire des groupes basés sur des tests plus fins. Ex : ((ou=100 ET cn=v) OU (ou=101 ET cn=w*)). Pour cela consulter la documentation en ligne (ou les commentaires de la présentation powerpoint).*

XII. Publication d'un canal :

Nous allons déployer un nouveau canal dans le portail afin de pouvoir le publier.

Décompresser l'archive esup-canal-hello-1.00.zip.

Adaptez le fichier build.properties (attention au sens des « / ») :

```
#Répertoire d'installation de Tomcat
tomcat.home = D:/uPortal_rel-2-4-3-quick-start/Tomcat_5-0-28
```

```
#Répertoire d'installation d'uPortal
```

```
uportal.home = D:/uPortal_rel-2-4-3-quick-start/uPortal_rel-2-4-3
```

```
#Répertoire de déploiement
```


```
deploy.home = D:/uPortal_rel-2-4-3-quick-start/Tomcat_5-0-28/webapps/uPortal
```

Faire un 'ant deploy' à la racine du canal (au besoin copier le fichier ant.bat dans le répertoire du canal)

Le canal déployé doit maintenant être publié afin d'être mis à la disposition des utilisateurs.

Publication à travers le portail :



Sur le portail cliquer sur  puis sur « Publish a new channel »

Choisir le type « Custom »

Faire pointer le canal vers la classe :

```
org.esupportail.portal.channels.CHelloWorld.CHelloWorld
```

Il n'y a pas de paramètres (faire « next »)

Cocher le « Has about » ce qui indique que le canal possède un fichier de description qui sera accessible à l'utilisateur.

Enfin choisir une ou plusieurs catégorie(s) et un ou plusieurs groupe(s) pour ce nouveau canal.

Terminer par « finished ».

Vous pouvez alors souscrire le canal comme on l'a vu ci-dessus. Vous devez pouvoir saisir un nom et afficher « Hello »

Retourner sur le gestionnaire de canaux et cliquer sur « Modify a currently published channel »

Supprimer la publication que vous venez de faire.

Remarque :

Si on se déconnecte et on se reconnecte le message « The <nom_channel> channel is no longer available. Please remove it from your layout ! » apparaît à la place du canal.

Publication par une directive ant :

Ouvrir le fichier CHello.xml et le modifier à votre convenance.

Copier le fichier CHello.xml dans le répertoire :

```
properties\chanpub du portail.
```

Remarque :

C'est dans ce répertoire qu'apparaissent les fichiers des canaux à publier par défaut.

Exécuter la tâche ANT 'deploy' pour prendre en compte les modifications et lancer la commande :

```
ant pubchan -Dchannel= CHello.xml
```


Relancer le portail.

Retourner sur le gestionnaire de canaux et cliquer sur « Modify a currently published channel ». Votre publication doit apparaître dans la liste.
Vous pouvez alors souscrire le canal comme on l'a vu ci-dessus.

XIII. Les Fragments :



Le portail permet la création d'ensembles définis de contenu associés à un ou plusieurs groupes. Ceci peut être très utile dans le cadre d'un déploiement au sein d'une université car les fragments permettent un affichage des canaux pertinents définis en fonction du public.

Fragments à travers le portail :

Accéder aux préférences par le bouton  et choisir « Fragments »
Créer un fragment pushed ou pulled

Modifier ensuite ses propriétés :

- y ajouter 2 canaux de la même manière que pour une souscription de canaux traditionnelle : « hello » et « Person Attributes » par exemple.
- Lier le fragment à un ou plusieurs groupe(s)

Remarque : Penser à enregistrer les préférences () avant de quitter () sinon elles seront perdues.

Déconnecter et reconnecter avec un utilisateur appartenant à un des groupes choisis.
Les fragments pushed apparaissent dans les onglets et les fragments pulled peuvent être souscrit de la même manière qu'un canal dans la catégorie « Fragments ».

Fragments par une directive ant :

Dans cette version seule la création de fragments pushed est possible (la nouvelle version de uPortal permettra cette opération).

Récupérer et adapter le fichier `frag.xml`.

Copier ce fichier dans :

`D:\esupdev\esupdev-2.4\Perso\uPortal\properties\al`

Si l'arborescence n'existe pas la créer (Le fichier sera recopié au moment du `esup.init` dans `D:\esupdev\esupdev-2.4\uPortal_rel-2-4-2-quick-start\uPortal_rel-2-4-2\properties\al`).

Exécuter la tâche ANT `'deploy'` pour prendre en compte les modifications.

Lancer la commande :

`ant pubfragments -DfragmentFile=properties/al/frag.xml`

Relancer le portail et se connecter avec un utilisateur appartenant à un des groupes choisis : le fragment apparaît.

Remarque : Ces fragments n'apparaissent pas dans le gestionnaire de fragments et ne peuvent donc pas être supprimés ou modifiés.