

3.6 Accès aux informations du portail

✓ Bon pour relecture

- cf "???" (phrase non terminée dans partie Utilisation)

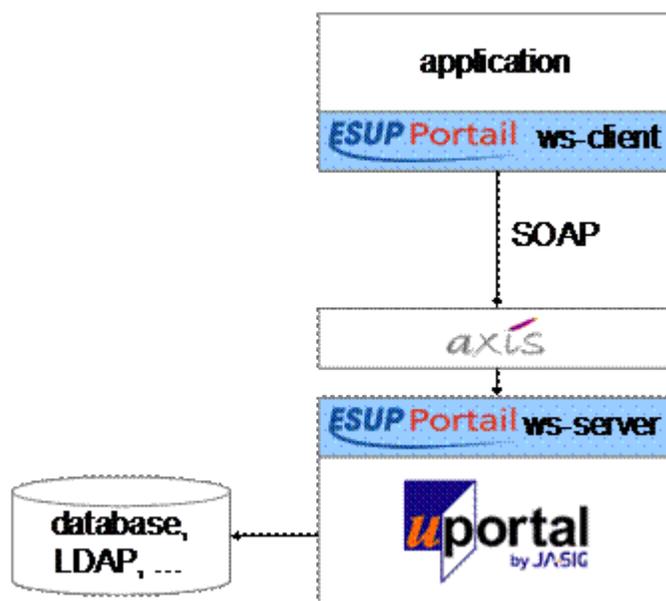
Sommaire :

- [Principe](#)
- [Utilisation](#)

Une des problématiques en mode *portlet* est liée au fait qu'une *portlet* s'exécute dans un contexte (*Tomcat*) différent de celui du portail. Il est ainsi impossible pour l'application d'accéder aux objets du portail, notamment pour récupérer les groupes et attributs utilisateurs (comme on le faisait avec de simples canaux *uPortal* (implémentations de l'interface **IChannel**)).

Principe

esup-commons s'appuie sur le projet *esup-portal-ws* (<http://www.ja-sig.org/wiki/display/UPC/esup-portal-ws+-a+web+service+to+publish+groups+and+user+attributes>) pour l'accès aux informations du portail. Celui-ci est basé sur un *web service* :



Le *JAR* de la partie cliente de *esup-portal-ws* est intégrée à *esup-commons*. L'application a donc à toutes les fonctionnalités de *esup-portal-ws*, pourvu que la partie serveur soit installée sur le portail auquel l'application va demander des informations.

Le lecteur se reportera à la documentation de *esup-portal-ws* pour savoir comment configurer la partie serveur.

Utilisation

Afin d'accéder aux groupes et attributs utilisateurs du portail, le développeur doit configurer le *bean portalService* dans le fichier de configuration *Spring Properties/portal/portal.xml*. Le *bean portalService* doit implémenter l'interface **PortalService**, qui possède les méthodes suivantes : ???

Une fois injecté (via *Spring*, par exemple dans un contrôleur), le *bean portalService* donne accès à toutes les informations voulues. Voici quelques exemples d'utilisation :

```

PortalService portalService = (PortalService) beanFactory.getBean("portalService");
PortalGroup rootGroup = portalService.getRootGroup();
PortalGroup group = portalService.getGroupById("local.10");
PortalUser user = portalService.getUser("paubry");
List<String> values = portalService.getUserAttributeValues(
    "paubry", "departmentNumber");
if (portalService.isUserMemberOfGroup("paubry", "local.10")) {
    ...
}

```

Voici enfin un exemple de configuration du **bean portalService**, en utilisant une implémentation qui offre des fonctionnalités de cache pour plus de performances :

```

<bean id="portalService"
    class="org.esupportail.portal.ws.client.support.uportal.CachingUportalServiceImpl" >
  <property name="url" value="http://localhost:8080/services/UportalService" />
  <property name="testUserId" value="paubry" />
  <property name="testGroupId" value="local.0" />
  <property name="testGroupName" value="Tous les groupes de personnes" />
  <property name="cacheManager" ref="cacheManager" />
</bean>

```

L'attribut **url** est l'*URL* sur laquelle le *web service* du portail est exposé, via *Axis*.

Les attributs **testUserId** et **testGroupId** servent à tester le *web service*, via la tâche *ant test-portal*.

Dans le cas où le développeur ne souhaite pas utiliser l'accès aux informations du portail, il utilisera la configuration suivante :

```

<bean id="portalService.servlet"
    class="org.esupportail.commons.services.portal.NotSupportedPortalServiceImpl" />

```

Dans ce cas, l'utilisation du **bean portalService** provoquera une exception.