

Tests UNR RUNN - Vincent bonamy

- [Authentification](#)
- [Groupes](#)
 - [JUEL ...](#)

Authentification

Configurations utilisées :

- Apache

```
RequestHeader set nuxeo-virtual-host "https://dsi-2.univ-rouen.fr/"
ProxyPass /nuxeo ajp://localhost:8009/nuxeo

<Location "/nuxeo">
    AuthType shibboleth
    ShibRequireSession Off
    ShibUseHeaders On
    Require shibboleth
</Location>
```

- nxserver/config/shibboleth-config.xml (en fait on a finalement privilégié l'usage d'un template : templates/runn/nxserver/config/shibboleth-config.xml)

```
<extension
  target="org.nuxeo.ecm.platform.shibboleth.service.ShibbolethAuthenticationService"
  point="config">
  <config>
    <uidHeaders>
      <!--uidHeader idpUrl="url1">uid1</uidHeader>
      <uidHeader idpUrl="url2">uid2</uidHeader-->
      <default>eppn</default>
    </uidHeaders>

    <loginURL>https://dsi-2.univ-rouen.fr/Shibboleth.sso/WAYF</loginURL>
    <logoutURL>https://dsi-2.univ-rouen.fr/Shibboleth.sso/Logout</logoutURL>

    <fieldMapping header="eppn">username</fieldMapping>
    <fieldMapping header="mail">email</fieldMapping>
  </config>
</extension>
```

-> l'authentification fonctionne.

Groupes

-> on déduit le fonctionnement de TestShibbolethComputedGroup.java (doc non trouvée)

Exemples qui fonctionnent :

```
true
currentUser.user.username == 'bonamvin@univ-rouen.fr'
```

Les **EL expressions** marchent bien ... mais cela ne semble définitivement **pas assez souple**, il nous faudrait quelque chose à base de **regexp** ou au moins il faudrait qu'on puisse faire des `endsWith`, `beginsWith`, `contains` etc. ...

En effet on va avoir notamment pour des attributs multi-valués type affiliation des choses comme :
affiliation=employee@unicaen.fr;member@unicaen.fr

JUEL ...

JUEL (<http://juel.sourceforge.net/>) est utilisé comme implémentation des Unified Expression Language (EL).

Précisément on retrouve la version 2.1.2 de juel (juel-impl-2.1.2.jar) dans le nuxeo nuxeo-dm-5.4.0-tomcat.

Les versions supérieures de JUEL (la dernière 2.2.2, par exemple) permettent nativement d'invoquer des méthodes dans les EL.

On propose donc d'utiliser cette nouvelle version :

- on supprime ./lib/el-api.jar et ./nxserver/lib/juel-impl-2.1.2.jar
- on ajoute directement ./lib/juel-2.2.2.jar

On peut alors utiliser pour la définition des groupes des définitions type 'foo'.matches('foo|bar') et même 'bonamvin@univ-rouen.fr'.equals(currentUser.user.username) ...

... pour autant currentUser.user.username.equals('bonamvin@univ-rouen.fr') ne fonctionne pas ou plutôt la validation de l'expression EL dans l'IHM ne fonctionne pas, on voit passer en effet dans les logs le message suivant :

```
2010-11-16 14:50:26,219 WARN [org.nuxeo.ecm.platform.el.DocumentModelResolver]
Property not found: user:username. Document 'null' with title 'null'
and type 'user' does not have any schema with prefix 'user'
```

... le document bidon servant à valider l'expression (validation == exécution de l'expression sur un document bidon) ne convient pas.

-> pour contourner le pb,

- on commente dans nuxeo-platform-login-shibboleth la validation effective de l'expression dans src/main/java/org/nuxeo/ecm/platform/shibboleth/computedgroups/ELGroupComputerHelper.java (patch [ci-joint](#))
- mvn package
- on récupère le jar ainsi créé nuxeo-platform-login-shibboleth-5.4.0.jar (donné [ci-joint](#)) pour le mettre à la place du jar initial dans ./nxserver/bundles/

On peut alors créer des définitions du type :

```
currentUser.user.username.matches('.*@univ-rouen.fr')
```

Attention cependant, il n'y a plus de validation d'effectuée sur vos expressions ...

à suivre ...