Configuration de shibboleth

Le consortium ESUP-Portail a financé un développement auprès de la société Nuxeo pour permettre l'utilisation de l'authentification shibboleth dans Nuxeo.

Vous pouvez vous référer à cette page pour plus d'informations sur le cahier des charges et la recette.

Pré-requis

Installation d'un SP provider : vous pouvez vous référer à la documentation disponible sur le site de la Fédération Education-Recherche

Avoir procédé à l'installation minimale de Nuxeo

Configuration de l'authentification Shibboleth

Récupérations des plugins

Deux plugins sont nécessaire pour l'authentification shibboleth. L'un permet l'authentification tandis que l'autre permet la définition et l'utilisation de groupes "shibboleth" (voir ci-dessous) et propose un affichage arborescent de groupes.

Vous pouvez récupérer les plugins d'origine sur le site de Nuxeo :

- nuxeo-platform-login-shibboleth : https://maven.nuxeo.org/nexus/content/groups/public/org/nuxeo/ecm/platform/nuxeo-platform-login-shibboleth/
- nuxeo-platform-shibboleth-groups-web : https://maven.nuxeo.org/nexus/content/groups/public/org/nuxeo/ecm/platform/nuxeo-platform-shibboleth-groups-web/

Choisissez et téléchargez la version qui correspond à votre instance de Nuxeo.

Nous proposons une version un peu modifiée par rapport à celle de base de Nuxeo. Cette dernière présente les différences suivantes :

- Ajout d'un paramètre permettant de spécifier une URL de retour après déconnexion du SP

- Supression du champ permettant de donner des droits à un utilisateur non connu (qui est créé dans Nuxeo).

- Modification des fichiers xhtml de présentation pour permettre de choisir l'ajout du mail dans l'auto-complétion lors de la recherche d'utilisateurs

Vous pouvez télécharger ces plugins ici (nécessite nuxeo 5.5) : nuxeo-platform-login-shibboleth et nuxeo-platfform-shibboleth-groups-web

Déposez les dans le répertoire templates/custom/bundles (ou dans un template spécifique shibboleth que vous n'oublierez pas d'inclure ensuite dans nuxeo.conf)

Configuration du fichier esup-login-config.xml

Éditez le fichier templates/custom/config/esup-login-config.xml (ou continuez à utiliser un template spécifique shibboleth) et modifiez le avec vos propres paramètres :

```
<component name="ecm.shibboleth.config">

<!-- certains composants doivent etre chargés avant lecture de ce fichier pour que cette configuration soit
prise en compte -->
<require>org.nuxeo.ecm.platform.ui.web.auth.defaultConfig</require>
<require>org.nuxeo.ecm.platform.ui.web.auth.WebEngineConfig</require>
<require>org.nuxeo.ecm.platform.usermanager.UserManagerImpl</require>

<!-- chainage d'authentification, on garde l'authentification BASIC pour les flux RSS/REST/cmis ...) -->
  <extension target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService" point="chain">
    <authenticationChain>
      <plugins>
        <plugin>BASIC_AUTH</plugin>
       <plugin>SHIB_AUTH</plugin>
        <plugin>ANONYMOUS_AUTH</plugin>
      </plugins>
    </authenticationChain>
  </extension>
```

```
<!-- configuration de l'authentification shibboleth -->
<extension target="org.nuxeo.ecm.platform.shibboleth.service.ShibbolethAuthenticationService"  point="config">
    <config>
      <uidHeaders>

        <!-- on peut specifier  l'attribut qui sert d'identifiant suivant l'idp, ici on choisit l'uid pour
notre idp local -->
        <uidHeader idpUrl="https://idp.my-univ.fr/idp/shibboleth">uid</uidHeader>

        <!-- pour tous les autres idp, on utilise l'eppn -->
        <default>eppn</default>
      </uidHeaders>

      <!--  URL de login du SP -->
      <loginURL>https://nuxeo.my-univ.fr/Shibboleth.sso/DS</loginURL>

      <!-- URL de logout du SP -->
      <logoutURL>https://nuxeo.my-univ.fr/Shibboleth.sso/Logout</logoutURL>

      <!-- URL de redirection une fois la deconnexion effectuee, uniquement avec plugin esup -->
     <logoutReturnURL>https//www.my-univ.fr/</logoutReturnURL>

      <!-- Mapping entre les attributs remontés via Shib et les attributs utilisateurs du schema user de nuxeo
-->
      <fieldMapping header="mail">email</fieldMapping>
      <fieldMapping header="givenName">firstName</fieldMapping>
      <fieldMapping header="sn">lastName</fieldMapping>
      <fieldMapping header="supannEtablissement">company</fieldMapping>
    </config>
  </extension>

  <!-- Gestion des utilisateurs admin/guest et des groupes virtuels -->
  <extension target="org.nuxeo.ecm.platform.usermanager.UserService"
    point="userManager">
   <userManager>
      <users>
        <anonymousUser id="Guest">
          <property name="firstName">Anonyme</property>
          <property name="lastName">Utilisateur</property>
           <property name="email">foo@bar.org</property>
       </anonymousUser>
      </users>
    <defaultAdministratorId>joe</defaultAdministratorId>

    <!-- en cas d'utilisation avec un annuaire, possibilite de choisir un groupe d'admins -->
    <administratorsGroup>myAdmingroup</administratorsGroup>
      <defaultGroup>members</defaultGroup>
    </userManager>
  </extension>

<!-- extension pour l'interface de gestion des groupes
     definit un separateur et un repertoire de base, utilise pour contruire l'arborescence des groupes (pour
une utilisation avec grouper) -->
<extension target="org.nuxeo.ecm.platform.shibboleth.web.service.ShibbolethGroupsService" point="config">
    <config>
      <parseString>:</parseString>
      <basePath>groupes:GroupesExternes</basePath>
    </config>
</extension>
</component>
```

Configuration du frontal apache

Il faut également configurer le frontal apache comme dans l'exemple donné ci-dessous (+ mod_jk ou mod_ajp):

```
ProxyPass /nuxeo ajp://localhost:8009/nuxeo
<Location "/nuxeo">
        AuthType shibboleth
        ShibRequestSetting requireSession 0
        ShibUseHeaders On
        Require shibboleth
</Location>
```

Les groupes "Shib"

Il est possible de définir des groupes en utilisant les attributs utilisateurs. Par exemple on peut construire un groupe de tous les utilisateurs d'un même établissement avec la définition suivante :

```
currentUser.user.company =='{UAI}03314764N'
```

PB : Pour pouvoir utiliser les expressions régulières et tester les attributs SHIB multivalués correctement, il est nécessaire d'utiliser une bibliothèque juel plus récente que celle livrée par défaut dans nuxeo.

Si vous souhaitez pouvoir utiliser cette fonctionnalité, il vous faudra donc :

Récupérez la dernière version de juel sur http://juel.sourceforge.net/,

Supprimez les bibltiohèques el-api.jar (<racine_nuxeo/lib>) et juel-impl-2.1.2.jar (<racine_nuxeo>/nxserver/lib) et placez la version récupérée de juel dans <racine_nuxeo>/lib.

Il conviendra aussi de modifier (nécessite de récupérer les sources) le fichier nuxeo-platform-login-shibboleth/src/main/java/org/nuxeo/ecm/platform /shibboleth/computedgroups/ELGroupComputerHelper.java pour commenter le test de validation de l'expression (cfpatch joint). refaites ensuite un package (mvn package) que vous placerez dans le répertoire bundles du template choisi (custom ou shibboleth).

A partir de là, il vous sera alors possible d'utiliser des expressions plus complexes comme celle-ci :

```
(currentUser.user.affiliation=='student' andcurrentUser.user.username.matches('^g')) or
(currentUser.user.username.contains('jaune')) and (not
(currentUser.user.email.matches('.*@etu.u-bordeaux1.fr') or
currentUser.user.company==('CNRS')))
```

Attention cependant, pour rendre cela fonctionnel, il est donc nécessaire de désactiver les vérifications de cohérences des expressions utilisées. Si jamais vous saisissez une définition de groupe inexacte, tout l'affichage des groupes ne fonctionnera plus et vous devrez aller dans la base de données pour supprimer le groupe dont l'expression comporte une erreur. A manier donc avec une grande précaution tant que l'implémentation n'est pas plus avancée.

Avec le plugin Shib, les groupes sont affichés de manière arborescente (le champ de saisie ne sert que pour retrouver les utilisateurs). Ceci est particulièrement intéressant pour les établissements qui utilisent grouper car cela permet de retrouver la même arborescence dans nuxeo.

⚠ Nous avons également porter cet affichage arborescent des groupes pour les établissements qui utiliseraient CAS/LDAP et grouper dans un plugin spécifique.

Utilisation du multi-Directory

Il peut s'avérer utile pour un établissement de proposer une authentification shibboleth d'une part mais également de pouvoir s'appuyer sur son annuaire local pour récupérer des informations concernant ses utilisateurs (et ses groupes) d'autre part. Pour cela il faut définir deux sources de données pour les utilisateurs et les groupes : une source locale à Nuxeo (dans laquelle seront stockés tous les groupes locaux, les groupes shib et les utilisateurs authentifiés via shib mais non présent dans l'annuaire) et une source correspondant à l'annuaire de l'établissement.

Configuration du fichier multi-directory-config.xml

Editez le fichier **templates/custom/config/multi-directory-config.xml** comme suit avec vos propres paramètres :

```
<component name="ecm.ldap.config">
```

```
<require>org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory</require>
<require>org.nuxeo.ecm.directory.sql.storage</require>

<extension target="org.nuxeo.ecm.directory.multi.MultiDirectoryFactory"
  point="directories">
  <directory name="userDirectory">

    <!-- definition des sources de donnees pour les utilisateurs -->
    <!-- schema utilise -->
    <schema>user</schema>

    <!-- attributs correspondant (dans le schema nuxeo) à l'identifiant et au mot de passe -->
    <idField>username</idField>
    <readOnly>false</readOnly>
    <passwordField>password</passwordField>

    <!-- déclaration de la source ldap, definie plus bas -->
    <source name="ldapUserDirectory">
      <subDirectory name="ldapUserDirectory"/>
    </source>

    <!-- declaration de la source locale que nous allons definir dans un autre point d'extension -->
    <source name="sqlUserDirectory" creation="true">
      <subDirectory name="sqlUserDirectory"/>
    </source>
  </directory>
</extension>

<extension target="org.nuxeo.ecm.directory.multi.MultiDirectoryFactory"
  point="directories">
  <directory name="groupDirectory">
    <!-- definition des sources de donnees pour les groupes -->

    <!-- schema utilise -->
    <schema>group</schema>

     <!-- attribut correspondant à l'identifiant du groupe (dans schema nuxeo) -->
    <idField>groupname</idField>
    <readOnly>false</readOnly>

    <!-- declaration de la source ldap pour les groupes, definie plus bas -->
    <source name="ldapGroupDirectory">

      <subDirectory name="ldapGroupDirectory"/>
    </source>

    <!-- declaration de la source locale definie dans un autre fichier -->
    <source name="sqlGroupDirectory" creation="true">
      <subDirectory name="sqlGroupDirectory"/>
    </source>
  </directory>
</extension>

<extension target="org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory"
  point="servers">

   <!-- definition de la connexion ldap au serveur ldap-->
  <server name="default">
    <ldapUrl>ldap://ldap.univ-fr:389</ldapUrl>
    <bindDn>cn=applis,ou=admin,dc=my-univ,dc=fr</bindDn>
    <bindPassword>verySecret</bindPassword>
  </server>
</extension>

<extension target="org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory"
  point="directories">

  <!--definition de l'annuaire comme source de donnees utilisateur -->
  <directory name="ldapUserDirectory">

    <!-- utilise la connexion qu'on vient de définir juste avant -->
```

```
      <server>default</server>
      <schema>user</schema>
      <idField>username</idField>
      <passwordField>password</passwordField>
      <searchBaseDn>ou=people,dc=my-univ,dc=fr</searchBaseDn>
      <searchClass>person</searchClass>
       <searchFilter>(&amp;(!(eduPersonAffiliation=affiliate))(status=valide))</searchFilter>
      <searchScope>onelevel</searchScope>
      <readOnly>true</readOnly>
      <cacheTimeout>3600</cacheTimeout>
      <cacheMaxSize>1000</cacheMaxSize>
      <querySizeLimit>0</querySizeLimit>
      <creationBaseDn>ou=people,dc=u-bordeaux1,dc=fr</creationBaseDn>
      <creationClass>top</creationClass>
      <creationClass>person</creationClass>
      <creationClass>organizationalPerson</creationClass>
      <creationClass>inetOrgPerson</creationClass>
      <rdnAttribute>uid</rdnAttribute>

      <!-- mapping d'attributs, doit être en cohérence avec ce qui est stipulé dans le fichier esup-login-
config.xml
           Dans notre exemple, on utilisait l'uid pour notre idp local, on fait donc un mapping avec l'uid) -->
      <fieldMapping name="username">uid</fieldMapping>
      <fieldMapping name="firstName">givenName</fieldMapping>
      <fieldMapping name="lastName">sn</fieldMapping>
      <fieldMapping name="company">supannEtablissement</fieldMapping>
      <fieldMapping name="email">mail</fieldMapping>
      <references>
        <inverseReference field="groups" directory="groupDirectory"
          dualReferenceField="members" />
      </references>
    </directory>
  </extension>

  <extension target="org.nuxeo.ecm.directory.ldap.LDAPDirectoryFactory"
    point="directories">

    <!-- on definit maintenant l'annuaire comme source de donnees -->
    <directory name="ldapGroupDirectory">

     <!-- connexion utilisee -->
      <server>default</server>

      <!-- schema utilise -->
      <schema>group</schema>

     <!--attribut correspondant à l'identifiant (dans schema nuxeo) -->
      <idField>groupname</idField>
      <searchBaseDn>ou=groups,dc=my-univ,dc=fr</searchBaseDn>
      <searchFilter>(&amp;(objectClass=groupOfNames)(cn=groupes:my-univ:Personnels:*))</searchFilter>
      <searchScope>subtree</searchScope>
      <cacheTimeout>3600</cacheTimeout>
      <cacheMaxSize>1000</cacheMaxSize>
      <querySizeLimit>10000</querySizeLimit>
      <creationBaseDn>ou=grouper,dc=u-bordeaux1,dc=fr</creationBaseDn>
      <creationClass>top</creationClass>
      <creationClass>groupOfUniqueNames</creationClass>
      <rdnAttribute>cn</rdnAttribute>

      <!-- Mapping entre attributs du schema nuxeo et attributs de l'annuaire -->
      <fieldMapping name="groupname">cn</fieldMapping>
      <references>
        <ldapReference field="members" directory="ldapUserDirectory"
          forceDnConsistencyCheck="false"
          staticAttributeId="member"
          dynamicAttributeId="memberURL" />
        <ldapReference field="subGroups" directory="ldapGroupDirectory"
          forceDnConsistencyCheck="false"
          staticAttributeId="uniqueMember"
          dynamicAttributeId="memberURL" />
        <inverseReference field="parentGroups"
```

```
          directory="groupDirectory" dualReferenceField="subGroups" />
      </references>
    </directory>
  </extension>

</component>
```

Modification de la source de donnée locale

Il nous reste maintenant à définir la source de données locales :

Éditez le fichier **templates/custom/config/default-sql-directories-bundle.xml** comme suit avec vos propres paramètres :

```
<?xml version="1.0"?>
<component name="org.nuxeo.ecm.directory.sql.storage">
  <implementation />
  <require>org.nuxeo.ecm.directory.sql.SQLDirectoryFactory</require>
  <extension target="org.nuxeo.ecm.directory.sql.SQLDirectoryFactory"
    point="directories">
   <!-- definition de la source de donnees locale pour les utilisateurs declaree dans le fichier precedent -->
    <directory name="sqlUserDirectory">
       <!-- schema utilise -->
      <schema>user</schema>
       <!-- data source -->
      <dataSource>jdbc/nxsqldirectory</dataSource>

      <table>users</table>
      <idField>username</idField>
      <passwordField>password</passwordField>
      <passwordHashAlgorithm>SSHA</passwordHashAlgorithm>
      <autoincrementIdField>false</autoincrementIdField>
      <dataFile>users.csv</dataFile>
      <createTablePolicy>on_missing_columns</createTablePolicy>
      <querySizeLimit>15</querySizeLimit>

      <references>
        <inverseReference field="groups" directory="sqlGroupDirectory"
          dualReferenceField="members" />
      </references>

    </directory>

    <directory name="sqlGroupDirectory">
     <!-- definition de la source de donnees locale pour les groupes declaree dans le fichier precedent -->
    <schema>group</schema>
      <dataSource>jdbc/nxsqldirectory</dataSource>
      <table>groups</table>
      <idField>groupname</idField>
      <dataFile>groups.csv</dataFile>
      <createTablePolicy>on_missing_columns</createTablePolicy>
      <autoincrementIdField>false</autoincrementIdField>

      <!-- Add 10 min cache to avoid refetching the groups during login -->
      <cacheTimeout>360</cacheTimeout>
      <cacheMaxSize>1000</cacheMaxSize>

      <references>
        <tableReference field="members" directory="userDirectory"
          table="user2group" sourceColumn="groupId" targetColumn="userId" schema="user2group"
          dataFile="user2group.csv" />
        <tableReference field="subGroups" directory="sqlGroupDirectory"
          table="group2group" sourceColumn="parentGroupId"
          targetColumn="childGroupId" schema="group2group" />
        <inverseReference field="parentGroups" directory="sqlGroupDirectory"
          dualReferenceField="subGroups" />
      </references>

    </directory>

  </extension>
</component>
```