

Acès automation via nuxeo-platform-login-portal-sso

Introduction

L'objectif ici était de tester un accès via automation à nuxeo en utilisant le plugin de login portal-sso.

Ceci dans la perspective d'utiliser cette méthode dans de futures portlets.



[esup-file-manager](#) utilise déjà ce plugin mais via un client CMIS et pas automation. On verra qu'il y a quelques subtilités.

Mise en œuvre

La première chose à faire est de configurer le plugin de login portal-sso.



Je conseille la lecture de cette [réponse](#) sur answers.nuxeo.com

Néanmoins ma documentation précise un point concernant la surcharge d'une chaîne d'authentification spécifique introduite avec automation.

Configuration

Création d'un template dev_sso



Le plugin portal-sso se trouve sous <https://maven-eu.nuxeo.org/nexus/content/repositories/public-releases/org/nuxeo/ecm/platform/nuxeo-platform-login-portal-sso>

Contenu :

```
templates/dev_sso/config/portal-auth-config.xml
templates/dev_sso/bundles/nuxeo-platform-login-portal-sso-5.4.1-HF11.jar
templates/dev_sso/nuxeo.defaults
```

Explications :

- templates/dev_sso/nuxeo.defaults est obligatoire mais est vide ici
- templates/dev_sso/bundles/nuxeo-platform-login-portal-sso-5.4.1-HF11.jar est le jar du plugin de login portal-sso.
- templates/dev_sso/config/portal-auth-config.xml est le fichier de configuration (Cf. ci-dessous)
- Le template est pris en compte en modifiant la propriété nuxeo.templates de bin/nuxeo.conf (nuxeo.templates=default,dev_sso)

Le fichier portal-auth-config.xml

```

<component name="MyAPP.postal_sso">
  <require>org.nuxeo.ecm.platform.ui.web.auth.defaultConfig</require>
  <require>org.nuxeo.ecm.platform.login.Portal</require>
  <!-- RB : on surcharge la conf de org.nuxeo.ecm.automation.server.auth.config qui définit une
specificAuthenticationChain sans PORTAL_AUTH -->
  <require>org.nuxeo.ecm.automation.server.auth.config</require>

  <extension
    target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
    point="authenticators">
    <authenticationPlugin
      name="PORTAL_AUTH">
      <loginModulePlugin>Trusting_LM</loginModulePlugin>
      <parameters>
        <!-- define here shared secret between the portal and Nuxeo server -->
        <parameter name="secret">toto</parameter>
        <parameter name="maxAge">3600</parameter>
      </parameters>
    </authenticationPlugin>
  </extension>

  <!-- RB : On surcharge ici la specificAuthenticationChain de org.nuxeo.ecm.automation.server.auth.config -->
  <extension
    target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
    point="specificChains">

    <specificAuthenticationChain name="Automation">
      <urlPatterns>
        <url>(.)/*automation.*</url>
      </urlPatterns>

      <replacementChain>
        <plugin>AUTOMATION_BASIC_AUTH</plugin>
        <plugin>PORTAL_AUTH</plugin>
        <plugin>ANONYMOUS_AUTH</plugin>
      </replacementChain>
    </specificAuthenticationChain>

  </extension>

</component>

```

Explications :

- Ne pas oublier de dépendre de **org.nuxeo.ecm.automation.server.auth.config** afin de pouvoir en surcharger la configuration
- Le paramètre **secret** sera utilisé dans la code java
- On ajoute **<plugin>PORTAL_AUTH</plugin>** dans la définition de la chaîne d'authentification automation



Si on utilise nuxeo drive, il faut ajouter **<plugin>TOKEN_AUTH</plugin>** après **<plugin>PORTAL_AUTH</plugin>**

Le code java

Cf. [NuxeoRESTWithSSOTest.java](#)

A partir de la version 5.4.2 du client automation on dispose d'un objet PortalSSOAuthInterceptor. Il suffit d'utiliser ce dernier lors de l'établissement de la connexion. On le construit en lui passant en paramètre le secret contenu dans **portal-auth-config.xml** et l'uid de l'utilisateur courant.

Voici un exemple de code :

```

HttpAutomationClient client = new HttpAutomationClient("http://localhost:8080/nuxeo/site/automation");
String secret = "toto";
String user = "test";
client.setRequestInterceptor(new PortalSSOAuthInterceptor(secret, user));
Session session = client.getSession();

```

Pour utiliser la version 5.4.2 du client automation il suffit d'ajouter cette dépendance de le pom.xml de votre projet maven :

```
<dependency>
  <groupId>org.nuxeo.ecm.automation</groupId>
  <artifactId>nuxeo-automation-client</artifactId>
  <version>5.4.2</version>
</dependency>
```

Sachant que pour trouver cette dépendance il faut référencer le dépôt maven de nuxeo :

```
<repositories>
  <repository>
    <id>nuxeo</id>
    <url>https://maven.nuxeo.org/nexus/content/repositories/public-releases</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
</repositories>
```