

# A résoudre

## A résoudre ...

### nuxeo-platform-webdav-server - 5.1.6

#### code http = 404 correspond à un objet inexistant ou non autorisé (devrait être 403)

Dans le plugin nuxeo-platform-webdav-server, la méthode permettant de résoudre les urls est CoreHelper.resolveVirtualPath. Cette méthode vérifie que chaque dossier/fichier constituant un chemin existe via la méthode du Nuxeo Core AbstractSession.exists. La méthode AbstractSession.exists renvoie false si l'objet n'existe pas **OU** si l'utilisateur n'a pas les droits de BROWSE sur l'objet. => l'utilisateur obtient un 404 pour les objets qu'il n'a pas le droit de consulter.

=> En fait possibilité de récupérer dans le plugin une exception DocumentSecurityException, on propose donc le patch suivant qui catch cette exception et renvoie un 403 dans les différentes méthodes Webdav.

[nuxeo-platform-webdav-server-5.1.4.diff](#)

#### via webdav, les dossiers mis à la poubelle sont visibles

Est-ce un problème ?

#### L'url d'accès direct à un espace de travail :

[http://nuxeo.univ-rennes1.fr/nuxeo/nxdoc/default/7e9e1abb-e1e8-4b23-bfe0-12a1bf2677c2/view\\_documents?tabId=&conversationId=0NXMAIN&conversationIsLongRunning=true&nbsp;](http://nuxeo.univ-rennes1.fr/nuxeo/nxdoc/default/7e9e1abb-e1e8-4b23-bfe0-12a1bf2677c2/view_documents?tabId=&conversationId=0NXMAIN&conversationIsLongRunning=true&nbsp;)

N'est pas très lisible !

Ca marche quand on est déjà authentifié

On arrive sur la page d'accueil apres authentication CAS

#### via webdav, lorsqu'on renomme les dossiers (leurs titres), les chemins ne sont pas modifiés

Est-ce un comportement souhaitable ?

#### dépôt et publication d'un ensemble de pages type "site web" (ressource complexe)

A noter que le rendu des pages semble fonctionner dans les workspaces via l'accès webdav depuis un navigateur. => il doit être possible d'obtenir la même chose dans les sections ... via un accès particulier (type webdav / resolver d'url via l'uid du proxy ...).

Dans les sections il faudrait pouvoir versionner (puis publier) un répertoire entier.

Nous avons tester le plugin nuxeo-core-folder-versioning (les configurations sont actuellement dans le trunk d'esup-ecm, mais elles sont désactivées), il permet effectivement de versionner un répertoire entier.

- PB1 : Cependant ni la restauration, ni la visualisation des versions ne semblent fonctionner : les contenus des répertoires ne semblent pas avoir été versionnés ...
- PB2 : Lorsqu'on tente de publier une telle version cependant, nous obtenons l'exception suivante :

```
....
Caused by: java.lang.ClassCastException: org.nuxeo.core.versioning.folder.
FolderishDocumentVersionDelegate
    at org.nuxeo.ecm.core.repository.jcr.JCRSession.createProxyForVersion(JCRSession.java:361)
    at org.nuxeo.ecm.core.api.AbstractSession.createProxy(AbstractSession.java:1925)
....
```

On a un peu de mal à comprendre l'intérêt de la classe FolderishDocumentVersionDelegate qui est un wrapper de la classe FolderishDocumentVersion ...

=> A étudier un peu plus.

==> WebEngine devrait nous permettre de répondre à nos attentes en ce qui concerne l'accès aux documents. => un resolver de documents (précisément de versions) sur WebEngine serait à réfléchir.

#### sql-storage: nom de l'auteur

Le nom de l'auteur correspond toujours au nom de la personne authentifiée => pb.

#### esup-ecm 3.1 - versions

La gestion des versions buguent un peu a priori : parfois la montée de version ne se fait pas.

Quand on redépose un document via le drag and drop, une nouvelle version est créée automatiquement (avec le même numéro de version que la version précédente). [[A remonter à Nuxeo](#)]

## authentification

### WEBDAV inaccessible en écriture : le module nuxeo-platform-login-anonymous prenant la main sur l'authentification

On a patché le plugin nuxeo-platform-login-anonymous (5.1.4) pour qu'il accepte en paramètres des SkipUrlRetrieveIdentity\_\* (urls qui ne seront ps traités par la procédure d'identification de nuxeo-platform-login-anonymous). On peut alors avoir quelque chose comme :

```
<extension
  target="org.nuxeo.ecm.platform.ui.web.auth.service.PluggableAuthenticationService"
  point="authenticators">
  <authenticationPlugin name="ANONYMOUS_AUTH" enabled="true"
    class="org.nuxeo.ecm.platform.ui.web.auth.anonymous.AnonymousAuthenticator">
    <parameters>
      <parameter name="SkipUrlRetrieveIdentity_Dav">dav</parameter>
    </parameters>
  </authenticationPlugin>
</extension>
```

Cf le patch suivant donc : [nuxeo-platform-login-anonymous-5.1.4.diff](#)

### WEBDAV inaccessible en lecture anonyme via l'interface WEB

Le patch ci dessus permet aux utilisateurs de se connecter sur le webdav via l'authentification BASIC, en contrepartie il ne permet malheureusement plus une authentification anonyme ...

=> ce n'est pas forcément dérangeant ici ?

### nuxeo-platform-login-anonymous (5.1.4) et nuxeo-platform-login-cas (5.1.4) : problème lors d'un cancel sur l'authentification CAS

Les 2 plugins marchent bien ensemble, mis à part que lorsque l'authentification CAS est annulé par l'utilisateur, celui-ci ne peut plus être identifié comme anonyme : l'authentification anonyme ne prenant plus la main.

CF la classe AnonymousAuthenticator de nuxeo-platform-login-anonymous, un attribut dans la session de l'utilisateur

BLOCK\_ANONYMOUS\_LOGIN\_KEY est fixé pour permettre à l'authentification CAS de prendre la main. Dans le contexte d'une annulation d'authentification CAS, cet attribut n'est malheureusement pas supprimé.

## Problème de la poubelle

Une poubelle par répertoire : difficile à gérer => faut-il que la poubelle soit une poubelle d'administrateur : permettrait à l'administrateur de pouvoir restaurer des fichiers/répertoires supprimés par les utilisateurs.

## Misc ...

Présentation fonctionnelle de l'implémentation esup-ecm / ori-oai: [présentation\\_Nuxeo\\_fonc.pdf](#)