

HEG-796-22-040

TP

Jan Krause-Bilvin

2022-05-11

Exercice 1 : hiérarchie OCFL

Comme indiqué dans le cours, Fedora organise ses AIP de la façon suivante:

```
hash := sha256( fedoraId )  
chemin = hash[0:3]/hash[3:6]/hash[6:9]/hash
```

Démarez Fedora commons (cf. TP 010 et 020).

Au moyen du script *id2storage.py*, localisez la ressource <http://localhost:8080/rest/records/acv/dossiers/D000003962>.

Pour ce faire, ouvrez un terminal Anaconda (*Anaconda prompt*), rendez-vous dans le répertoire où vous avez dézippé Fedora Commons et exécutez:

```
python id2storage.py records/acv/dossiers/D000003962
```

Vous obtenez le chemin vers l'AIP contenant ce dossier relativement à la racine OCFL. Celui ci est indiqué par "Storage path".

Pour localiser ce dossier commencez par ouvrir la racine OCFL situé dans le sous répertoire *fcrepo-home/data/ocfl-root*. Suivez ensuite le chemin indiqué par le script.

La hiérarchie de stockage que vous venez de parcourir a pour but de répartir les AIP dans différents répertoires de manière à éviter des problèmes de surcharge quelque soit le système de stockage utilisé.

Rendu: capture d'écran de l'explorateur windows affichant le dossier de l'AIP (paquet OCFL) correspondant. Nom du fichier tp4ex1.png.

Exercice 2 : inventaire, versions et manifeste OCFL

Depuis le répertoire localisé précédemment, accédez à la version 3, répertoire *v3*, du dossier, et ouvrez le fichier *inventory.json* dans Firefox.

Retrouver la version 3 du fichier *fcr-container.nt* contenant les métadonnées du dossier (i.e. le RDF en format NT une extension de turtle).

Pour ce faire:

1. Localiser la section “version” et la sous section “v3” dans l’inventaire.
2. Localiser le fichier “fcr-container.nt” dans cette section “v3”.
3. Copier ou “noter” le checksum correspondant à ce fichier.
4. Dans la section “manifest” en haut du fichier manifest.json, sur la base du checksum copié, trouver le chemin de stockage du fichier “fcr-container.nt” recherché.

Vous constaterez qu’un fichier déposé déjà enregistré dans une version précédente d’un AIP OCFL n’est pas dupliqué dans les versions ultérieurs, même si il en fait toujours partie.

En résumé, cette structure permet: * de stocker toutes les versions d’un AIP * éviter les doublons de fichier au sein d’un AIP (économie d’espace de stockage) * d’associer le checksum à chaque fichier (métadonnées et données) et ainsi de pouvoir vérifier son intégrité

Rendu: déposer la version v3 fcr-container.nt sur cyberlearn en la nommant tp4ex2.nt .

Exercice 3 : unité d’archivage

Dans l’interface web de Fedora Commons, se rendre à l’adresse <http://localhost:8080/records/acv/dossiers/>

Puis:

- Y créer un nouveau dossier, i.e. “BasicContainer” avec l’id “conteneur”.
- De la même manière que dans l’exercice 1 du TP2, utiliser RiC pour (rico:title) pour lui attribuer le titre “Container basique.” ainsi que la description (rico:scopeAndContent): “Container LDP basique.”
- Comme dans l’exercice 3 du TP2, ajouter un nouveau fichier binaire (par exemple une image de votre choix) dans ce container.

Revenir dans <http://localhost:8080/records/acv/dossiers/> .

Cette fois:

- Créer un conteneur de type “Archival group” avec l’id “aip”.
- Lui donner le titre “Container archivistique.” ainsi que la description (rico:scopeAndContent): “Container de type archival unit.”
- Ajouter le même fichier binaire à ce nouveau conteneur.

Au moyen du script *id2storage.py* utilisé dans l’exercice 1, localisez et comparez les deux OFCL correspondants à:

- [records/acv/dossiers/conteneur](#)
- [records/acv/dossiers/aip](#)

Quelle est la principale différence entre les deux?

Rendu: déposer votre réponse rédigée sur cyberlearn en la nommant `tp4ex3.txt` .