Publication des shape d’un dossier

# Phase alimentation manuelle :

L’API REST comprenant un bug qui empêche la création d’un datastore geoserver de type ‘Directory of spatial files (shapefiles)’ il a été décidé que ce datastore serait créé manuellement par le biais de l’interface d’administration de Geoserver. Cette première phase comporte donc 2 étapes :

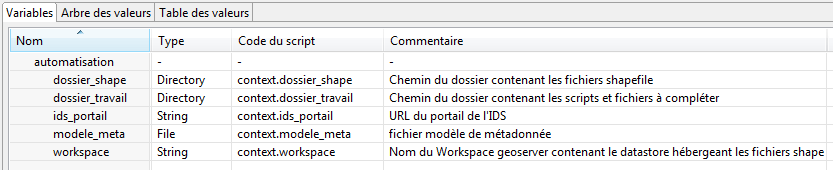
* Création du datastore de type ‘Directory of spatial files (shapefiles)’ sous geoserver.
* Alimentation du dossier par copie des fichiers shape via un outil tel que BitKinex ou une future version de Pydio. L’ensemble de ces fichiers doit être accessible à l’utilisateur qui exécute le processus tomcat.

# Préparation du fichier à compléter :

Afin d’automatiser le processus de publication des cartes et métadonnées associées, un job Talend va analyser le contenu du dossier préalablement alimenté pour fournir un listing des futures couches qu’il faudra ensuite compléter. Il fournit également un fichier par couche pour compléter les métadonnées de la couche.

Ce job est nommé **shape\_directory\_preparation\_listing**

Il convient de personnaliser les variables de contexte avant le lancement :



# Complément des fichiers :

L’utilisateur complète le listing des couches (informations pour geoserver ) et chaque fichier représentant la fiche métadonnée d’une couche ()informations pour geonetwork).

Le fichier prepa\_geoserver.csv comporte toutes les informations qui seront passées au script de création de couche :

|  |  |
| --- | --- |
| **Colonne** | **Description** |
| Nomfichier | Nom du fichier shape sans l’extension et en respectant la casse. |
| Ficsld | Nom du fichier style (sans l’extension et en respectant la casse) ou bien la chaîne ‘null’ |
| layer\_titre | Titre libre de la couche (sans caractère ‘ ;’) |
| layer\_unite | Unité de la couche (m par défaut) |
| layer\_EPSG | Code numérique EPSG des données de la couche. |
| layer\_resume | Description libre de la couche (sans caractère ‘ ;’) |
| layer\_keyword | Mot(s) clé(s) libre de la couche (sans caractère ‘ ;’) |
| ecrase\_style | Booléen contenant 1 si on veut forcer le remplacement d’un style en cas d’existence ou bien 0 (faux) si on veut laisser un style préexistant sans modification. |
| style\_global | Booléen contenant 0 (faux) pour indiquer que le style n’est pas global (donc associé au workspace) et 1 dans le cas contraire. |
| delete\_layer | Booléen contenant la valeur 1 pour forcer la suppression préalable de la couche si elle préexiste et 0 (faux) dans le cas contraire. |
| uuid\_metadata | Identifiant unique de la fiche de métadonnées. En principe il n’y a pas à retoucher ce qui a été généré. |

Chaque fichier portant le même nom que le fichier shape avec une extension .meta.xlsx représente les informations de métadonnées de la couche qui seront utilisées pour paramétrer le script de création de la métadonnée ; le mécanisme s’appuie sur le nommage des cellules d’un classeur Excel:

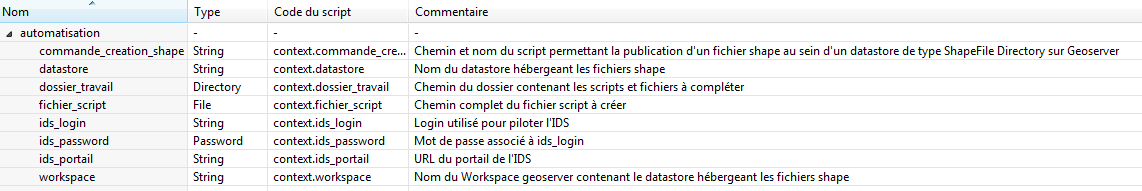
|  |  |
| --- | --- |
| **Cellule nommée** | **Description** |
| gmd\_title |  |
| gmd\_abstract |  |
| gmd\_hierarchyLevel |  |
| gmd\_purpose |  |
|  |  |
| ……….. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Préparation du script de création des couches et fiches de métadonnées.

Afin de générer les scripts de publication des couches et métadonnées un job talend va s’appuyer sur les fichiers complétés pour fournir un script permettant la publication des diverses couches et un script permettant la publication des fiches de métadonnées associées.

Ce job est nommé **shape\_directory\_script\_cartes**

Il convient de personnaliser les variables de contexte avant le lancement :



Le job fournit en sortie un script (selon la variable de contexte fichier\_script) qu’il conviendra de récupérer avec l’ensemble des scripts de manipulation de geoserver. Le tout devra être mis en place sur une machine linux, rendu exécutable.

Lancer creer\_couches.sh

Attendre 2-3 minutes ou faire un reload du geoserver pour etre sur que les couches sont chargées

Lancer creer\_meta.sh