

Automatisation du processus de publication vers une IDS

Présentation d'une méthode de travail

Jean-Baptiste Paroissien, Florent Millet, Nicolas Saby, Benoît
Toutain

INRA-InfoSol

6 Novembre 2014, réunion CATI-SIOEA

Les enjeux

Techniques

- Automatisation du processus de diffusion : développement de scripts
- Partage du code source : utilisation d'une forge de développement
- Continuité et pérennité du projet : documentation
 - ▶ Procédures
 - ▶ Modes opératoires
 - ▶ Documentation du code source
- **Traçabilité et reproductibilité des processus de diffusion** : fichiers de suivis

Les enjeux

Techniques

- Automatisation du processus de diffusion : développement de scripts
- Partage du code source : utilisation d'une forge de développement
- Continuité et pérennité du projet : documentation
 - ▶ Procédures
 - ▶ Modes opératoires
 - ▶ Documentation du code source
- **Traçabilité et reproductibilité des processus de diffusion** : fichiers de suivis

Respect des recommandations INSPIRE

- Interopérabilité de la diffusion des données
- Gestion des fiches de métadonnées au format INSPIRE

L'architecture fonctionnelle de diffusion

Centralisation de la documentation et du code source

Fonctions

- Construction fichiers de style (.sld, .fil)
- Publication vers Geoserver/GeoNetwork
- Construction fiches INSPIRE (.xml)

Documentation

- Procédure
- Modes opératoires
- Fichiers de suivis
- Documentation fonctions

Catalogage

- Inventaire des données à diffuser
- Caractéristiques sémiologiques
- Caractéristiques techniques de diffusion
- Données pour construire les métadonnées

L'architecture fonctionnelle de diffusion

Centralisation de la documentation et du code source

Fonctions

- Construction fichiers de style (.sld, .fil)
- Publication vers Geoserver/GeoNetwork
- Construction fiches INSPIRE (.xml)

Documentation

- Procédure
- Modes opératoires
- Fichiers de suivis
- Documentation fonctions

Catalogage

- Inventaire des données à diffuser
- Caractéristiques sémiologiques
- Caractéristiques techniques de diffusion
- Données pour construire les métadonnées

Base de données

Entrepôt



L'architecture fonctionnelle de diffusion

Centralisation de la documentation et du code source

Fonctions

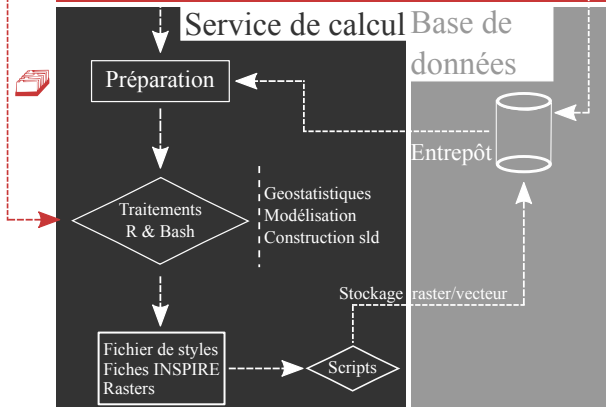
- Construction fichiers de style (.sld, .fil)
- Publication vers Geoserver/GeoNetwork
- Construction fiches INSPIRE (.xml)

Documentation

- Procédure
- Modes opératoires
- Fichiers de suivis
- Documentation fonctions

Catalogage

- Inventaire des données à diffuser
- Caractéristiques sémiologiques
- Caractéristiques techniques de diffusion
- Données pour construire les métadonnées



L'architecture fonctionnelle de diffusion

Centralisation de la documentation et du code source

Fonctions

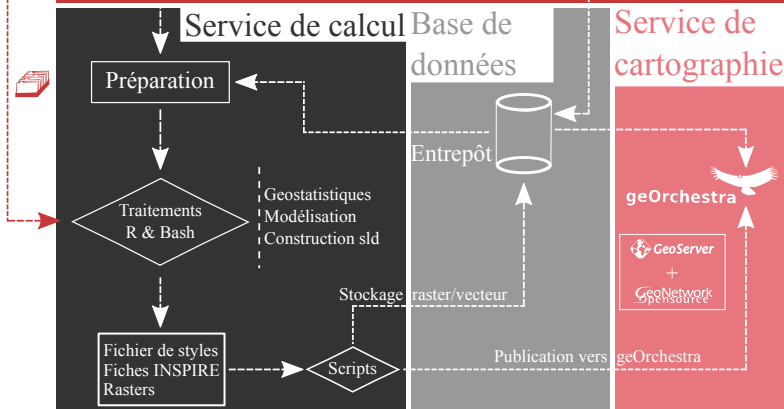
- Construction fichiers de style (.sld, .fil)
- Publication vers Geoserver/GeoNetwork
- Construction fiches INSPIRE (.xml)

Documentation

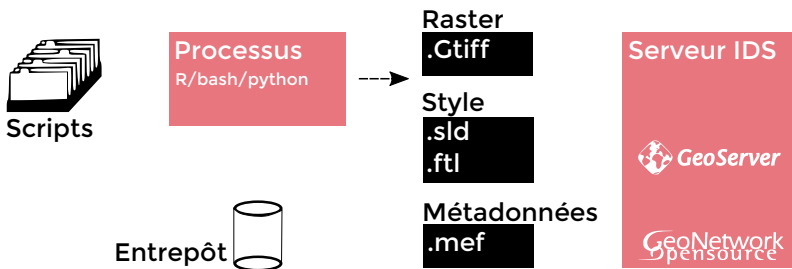
- Procédure
- Modes opératoires
- Fichiers de suivis
- Documentation fonctions

Catalogage

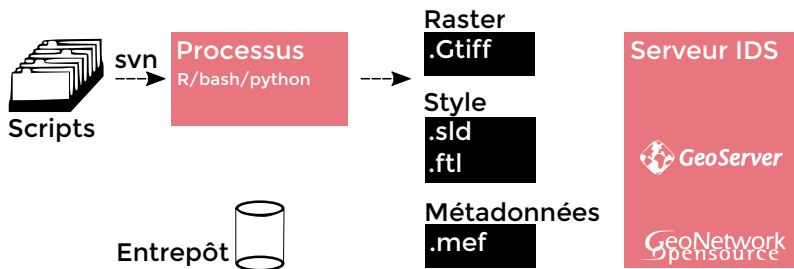
- Inventaire des données à diffuser
- Caractéristiques sémiologiques
- Caractéristiques techniques de diffusion
- Données pour construire les métadonnées



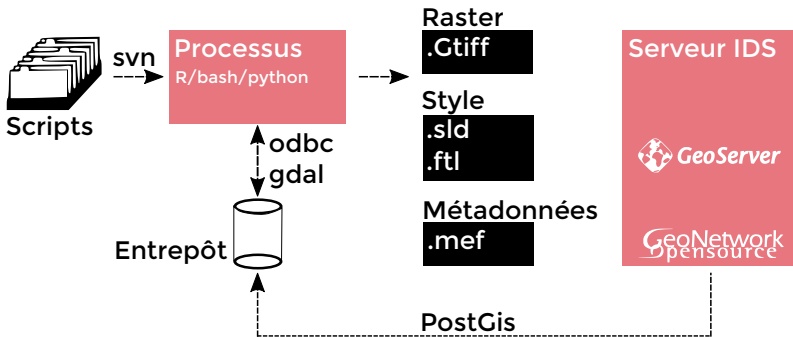
Interaction entre les services



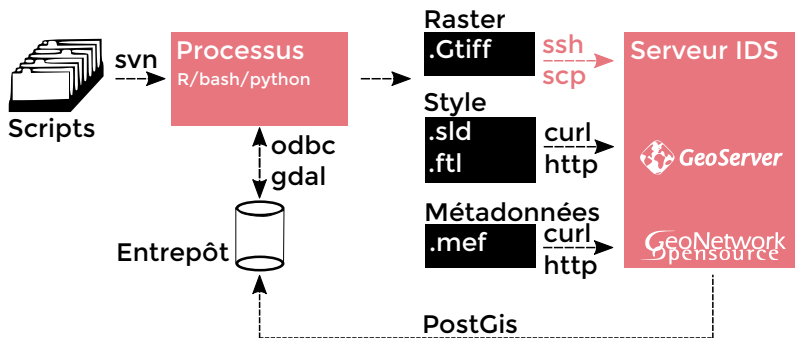
Interaction entre les services



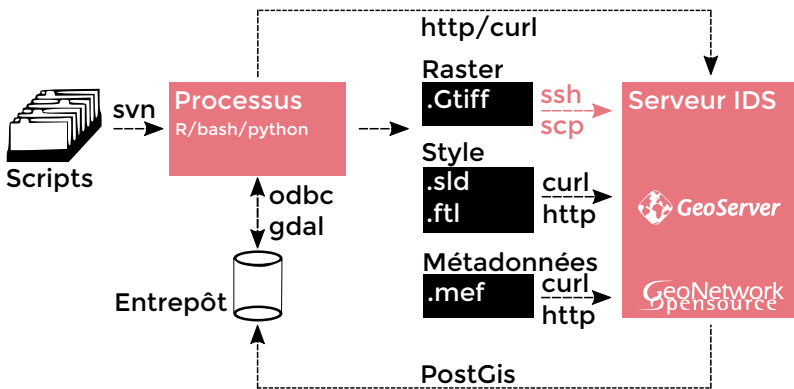
Interaction entre les services



Interaction entre les services



Interaction entre les services



Les différents langages

R+Markdown

- Fichiers de suivis
(**traçabilité/reproductibilité**)
- Création des données élaborées et fichiers de styles (.sld)

```
COMMENT ON COLUMN meta.webservices.descript_meta.resume IS 'Résumé de la donnée diffusée';
COMMENT ON COLUMN meta.webservices.descript_meta.geneologie IS 'Etat de l''historique du traitement et/ou de la qu
élite générale de la série de données géographiques.';
```

```
## caracterisi
```

5 Construction finale de la table catalogue

La construction de la table catalogue est finalisée avec la fonction R [catalogage](#).

```
## Chargement de la fonction catalogage.R
source("~/home/jb/Bureau/Scripts/svn/dev/equipe_traitement/webservices/trunk/fonctions/R/catalogage/catalogage.R")
repertoire <- "~/home/jb/Bureau/Scripts/svn/dev/equipe_traitement/webservices/trunk/Fichiers_suivis/Metawebservices
/Catalogage/Tables"
table_catalogue <- "catalogage"
```

```
## Lancement de la fonction
catalogage(repertoire,table_catalogue)
```

```
## [1] "Lecture des différentes tables"
## [1] "Finalisation de la création de la table"
## [1] "Enregistrement de la table"
## [1] "Ajout du titre, du résumé et de la généalogie"
## [1] "Mise à jour de la table"
## [1] "FIN"
```

```
## Ajout de commentaires sur la table et les champs de la table
sqlQuery(dela,"COMMENT ON TABLE meta.webservices.catalogage IS 'Table recensant l''ensemble des données diffusées
dans le cadre du projet webservice. Ces données regroupent à la fois les caractéristiques sémantiques, les caract
éristiques techniques de diffusion (pour le geoserver) et les éléments pour construire une fiche de métadonnée au
format INSPIRE. Le fichier de suivi de la création de cette table est consultable sur la forge : https://forge.or
leams.inra.fr/projects/webservice/repository/raw/trunk/Fichiers_suivis/Metawebservices/Catalogage/Suivis/PS_catalo
page.html");
COMMENT ON COLUMN meta.webservices.catalogage.programme IS 'Nom du programme d''acquisition de la donnée à diffuse
r (BIRDG; Metel);
COMMENT ON COLUMN meta.webservices.catalogage.campagne IS 'Numéro de la campagne d''acquisition (pour le moment, 1
)';
COMMENT ON COLUMN meta.webservices.catalogage.profondeur IS 'Profondeur de sol des analyses (pour le moment, 0-30)
';
COMMENT ON COLUMN meta.webservices.catalogage.type IS 'Type de données diffusée (rast pour raster ou vect pour vec
teur)';
COMMENT ON COLUMN meta.webservices.catalogage.id_element IS 'Nom de champs attributaire dans Donesol3';
COMMENT ON COLUMN meta.webservices.catalogage.securite IS 'Type de source (source pour acquisition, stack pour depar
```

Les différents langages

R+Markdown

- Fichiers de suivis
(traçabilité/reproductibilité)
- Création des données élaborées et fichiers de styles (.sld)

bash

- Transfert fti/Gtiff vers IDS (ssh/scp)
- Administration geoserver (curl)

```

ftoinspire.sh (~/.Bureau/Scripts/svn/dev/.../webservices/trunk/Fonctions/bash) - GVIM3
Fichier Edition Outils Syntaxe Buffers Fenêtre Plugin Help Aide

1 #!/bin/bash
2 # **
3 # *
4 # * Titre ftoinspire
5 # * Aauthor Jean-Baptiste Paroissien
6 # * #description Création d'une fiche de métadonnée au format INSPIRE
7 # * #params titre
8 # * #params resume
9 # * #params ddate debut
10 # * #params ddate fin
11 # * #params keywords
12 # * #params stypedonnees
13 # * #params idurl
14 # * #params repertoire Chemin du répertoire où sera enregistré la fiche (Format : "/xxx/xxx/")
15 # * #params adresse URL de l'IDS
16 # * #params workspace Nom du workspace de geoserver
17 # * #params couche Nom de la couche à visualiser (correspond au champs 'couche' de la table cat
18 # * #params uuid Universally Unique Identifier : Identifier aléatoire généré avec le paquet R
19 # * #params path_python Répertoire de stockage des fonctions python
20 # * #params genealogie argument pour la généalogie de la ressource publiée
21
22 # * #return La fonction crée un fichier XML, un aperçu png du webservice et convertie l'ensem
23 # * e dans un fichier zip de métadonnées (format MEF). Le fichier est stocké dans le répertoire pré
24 # * paré à cet effet (argument repertoire)
25
26 # *
27 # *
28 # * Variables variable
29
30 titre=${1}
31 resume=${2}
32 ddate_debut=${3}
33 date_fin=${4}
34 keywords=${5}
35 typedonnees=${6}
36 idurl=${7}
37 repertoire=${8}
38 adresse=${9}
39 workspace=${10}
40 couche=${11}
41 uuid=${12}
42 path_python=${13}
43 genealogie=${14}
44
45 # Variable fixe la priori
46 ca_organisme="GIS Sol"
47 ca_adresse="2163 avenue de la pomme de pin"
48 ca_ville="Orléans"
49 ca_codepostale="45075"

```

Les différents langages

R+Markdown

- Fichiers de suivis
(traçabilité/reproductibilité)
- Création des données élaborées et fichiers de styles (.sld)

bash

- Transfert ftp/Gtiff vers IDS
(ssh/scp)
- Administration geoserver
(curl)

python

- Création métadonnées
(.mef)

```

xml_to_mef.py (-/Bureau/Scripts/svn/dev/...eservices/trunk/Fonctions/python) - GVIM4
Fichier  Edition  Outils  Syntaxe  Buffers  Fenêtre  Plugin  Help  Aide

2 # This is gn iso19139 to_mef example.py
3
4 ***
5 Example script to create GeoNetwork's Metadata Exchange Format 1.1 (MEF) archive from ISO 19139
  metadata XML files
6
7 MEF files are ZIP archives with the following structure:
8 +-+---->.mef      zip archive with the metadata's UUID (must be valid) as the file name
9 | -info.xml       GeoNetwork (GN) specific metadata such as privileges, related data & thumbna
10 | -metadata.xml  ISO 19139 metadata record
11 |   +public      directory with public thumbnail and data files - can be empty
12 |   +private     directory with private (GN authentication required) data files such as shape
  files etc. - can be empty
13
14 Usage: make sure to edit mef_siteId, info_xml, etc.
15
16 Arg 1 : ISO metadata folder path
17 Arg 2 : MEF metadata folder path
18 Arg 3 : Temporary workspace to place MEF content before zipping up
19 Arg 4 : Temporary directory that is created and deleted
20
21 Example : python gn iso19139 to_mef.py '/home/jb/Bureau/ficheinspire/xml/' '/home/jb/Bureau/fich
  einspire/mef/' '/home/jb/Bureau/ficheinspire/tmp/' 'temp_mef/'
22
23 Python 2.6
24 Wolfgang Grunberg
25 Arizona Geological Survey
26 11/09/2009
27 ***
28
29 # Library Imports - not all may be needed
30 import os
31 import sys
32 import cProfile
33 import shutil
34 import zipfile
35 import mimetypes
36 from xml.dom import minidom
37 from xml.dom import Node
38 from time import strftime
39
40 # module globals and constants
41 # globals
42 _author_ = "Wolfgang Grunberg"
43 _copyright_ = "Copyright 2009, Arizona Geological Survey"
44 _credits_ = ["Wolfgang Grunberg", "the Internets"]
45 _license_ = "GPL"
46 _version_ = "1.0.0"
47 _maintainer_ = "Wolfgang Grunberg"
48 _email_ = "wgrunberg@azgs.az.gov"
python/xml_to_mef.py [FORMAT=unix] [TYPE=PYTHON] [ASCII=000] [HEX=00] [POS=0015,0001] [5$] [LEN=240]

```

La table catalogage

Administration geoserver

- Nom workspace
- Nom entrepôt
- Nom et titre couche, ...

La table catalogage

Administration geoserver

- Nom workspace
- Nom entrepôt
- Nom et titre couche, ...

Fiches métadonnées

- Titre
- Résumé
- Généalogie, mots clés
- uuid

La table catalogage

Administration geoserver

- Nom workspace
- Nom entrepôt
- Nom et titre couche, ...

Fiches métadonnées

- Titre
- Résumé
- Généalogie, mots clés
- uuid

Fichiers de style

- Palette couleur
- Classification
- unité, ...

Exemple d'application

```
223 ## Diffusion dans GeoServer
224
225 ```{r, tidy=FALSE,eval=TRUE}
226 for(i in 1:nrow(catalogage)) {
227   couche <- catalogage[i,"couche"]
228   titre <- catalogage[i,"titre"]
229   resume <- catalogage[i,"resume"]
230   unite <- catalogage[i,"unite"]
231   keyword <- catalogage[i,"mots_cles"]
232   nomsld <- catalogage[i,"id_element"]
233   EPSG <- catalogage[i,"epsg"]
234   repstyle <- "/data/cartes/gissol/style/"
235   repftl <- paste("/data/work/gis_data/workspaces/",workspace,"/",entrepot,"/",couche,sep="")
236   rep <- "/data/cartes/gissol/style/"
237   replocal <- repertoire
238   serveur <- "147.99.222.233"
239   login_serveur <- "root"
240
241   # Construction de la liste des arguments
242   argpubli <- paste(workspace,entrepot,couche,nomsld,titre,unite,EPSG,resume,keyword,adresse,
login_ids,mdp_ids,sep="\ " "\")
243   argpubli <- paste("\",argpubli,\"\",sep="")
244
245   # Publication du vecteur PostGIS dans le geoserver
246   system(paste("sh ",repbash,"vtogeoserver.sh ",argpubli,sep=""))
247
248   # Envoie du ftl vers le serveur de l'IDS (commandes ssh)
249   system(paste("sh ",repbash,"ftltogeoserver.sh ",argsld,sep=""))
250 }
251
252 ```
```

Publication d'un raster vers un GeoServer, 1

Préparation du répertoire

```
# Creation repertoire
ssh ${login_serveur}@${serveur} "mkdir -p /data/cartes/\
${workspace}/${programme}/raster_${couche}"

# Copie raster
scp ${replocal}${nomraster} ${login_serveur}@${serveur}:/data/cartes/\
${workspace}/${programme}/raster_${couche}/

# Suppression couche, style, entrepot
curl -u ${login_ids}:${mdp_ids} -XDELETE -H 'Accept:text/xml'\
${adresse}/geoserver/rest/layers/${couche}

curl -v -u ${login_ids}:${mdp_ids} -d "purge=true" -X DELETE \
${adresse}/geoserver/rest/styles/${nomsld}

curl -u ${login_ids}:${mdp_ids} -XDELETE "${adresse}/geoserver/rest/\
workspaces/${workspace}/coveragestores/${entrepot}?recurse=true"
```

Publication d'un raster vers un GeoServer, 2

Publication du raster

```
# Ajout du raster
curl -u ${login_ids}:${mdp_ids} -XPOST -H 'Content-type: text/xml' \
-d "<coverageStore><name>${couche}</name><enabled>true</enabled> \
<type>GeoTIFF</type><url>file:/data/cartes/${workspace}/${programme}/ \
raster_${couche}/${nomraster}</url><workspace><name>${workspace}</name> \
</workspace></coverageStore>" \
${adresse}/geoserver/rest/workspaces/${workspace}/coveragestores?configure=all

# Configuration (type projection, titre, resume de la couche...)
curl -u ${login_ids}:${mdp_ids} -XPOST -H 'Content-type: application/xml' \
-d "<coverage><name>${couche}</name><enabled>true</enabled><title> \
${titre}</title><abstract>${resume}</abstract><keywords><string>${keyword}</string>\
</string></keywords><projectionPolicy>REPROJECT_TO_DECLARED\
</projectionPolicy><srs>EPSG:${EPSG}</srs><nativeCRS>EPSG:${EPSG}</nativeCRS>\
</coverage>"${adresse}/geoserver/rest/workspaces/${workspace}/coveragestores/\
${entrepot}/coverages

# Configuration du fichier de style préalablement copie
curl -u ${login_ids}:${mdp_ids} -XPOST -H 'Content-type: text/xml' -d \
"<style><name>${nomsld}</name><filename>${repstyle}${nomsld}.sld</filename>\
</style>" ${adresse}/geoserver/rest/styles

# Chargement du style
curl -u ${login_ids}:${mdp_ids} -XPUT -H 'Content-type: \
application/vnd.ogc.sld+xml' -d @"${nomsld}.sld" ${adresse}/\
geoserver/rest/styles/${nomsld}
```

Documentation du projet

Procédures

- Pdf reliés à des modes opératoires
- Documentation versionnée dans le dépôt svn

The screenshot shows a web browser window with the address bar displaying 'https://forge.orteparis.inra.fr/projects/webseminet/wiki'. The page title is 'Traitement-Infosol - webservice'. The navigation menu includes 'Accueil', 'Articles', 'Demandes', 'Nouvelle demande', 'Carte', 'Calendrier', 'Annuaire', 'Demandes', 'Wiki', 'Fichiers', 'Dépôt', and 'Configurations'. The main content area is titled 'Les Webservices cartographiques du GISSOL : administration, production et utilisation'. It contains a paragraph of text, a bulleted list of links, and a paragraph of text. The footer of the page reads 'Powered by Sphinx © 2006-2012 Jean-Philippe Lemaire'.

https://forge.orteparis.inra.fr/projects/webseminet/wiki

traitement-Infosol - webservice

Accueil Accueil Demandes Nouvelle demande Carte Calendrier Annuaire Demandes Wiki Fichiers Dépôt Configurations

Les Webservices cartographiques du GISSOL :
administration, production et utilisation

Ce wiki présente l'ensemble de la documentation nécessaire à l'administration, l'implémentation et l'utilisation des webservices cartographiques du GISSOL. Il s'adresse donc à l'ingénieur quel que type d'utilisateur souhaitant :

- Produire des données géométriques et géographiques dans une infrastructure de données spatiales
- Utiliser des webservices dans un système d'information géographique ou une application web
- Administrer l'ensemble de l'architecture fonctionnelle

Une page proposée en base sur des fichiers pdf qui oriente le lecteur vers différents modes opératoires (en ligne). Nous vous conseillons de bien lire l'ensemble du contenu de ces documents afin d'appréhender le fonctionnement et l'architecture de travail. Trois éléments d'orientation sont proposés, adaptés aux différents types d'utilisateurs :

- Les administrateurs : /:doc
- Les producteurs : /:doc
- Les utilisateurs : /:doc

Si vous avez des remarques, n'hésitez pas à nous contacter directement : jean-philippe.lemaitre@orteparis.inra.fr ou à utiliser les fonctionnalités de la forge.

Nouveaux Fichiers

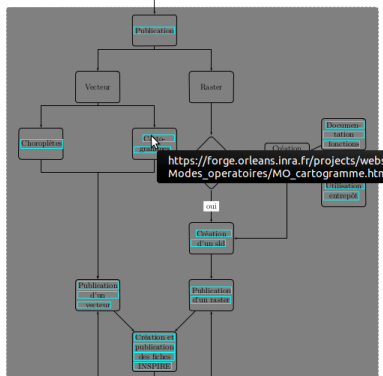
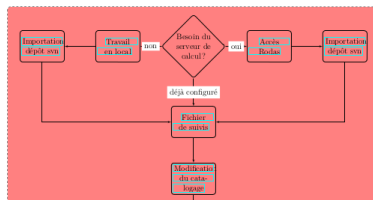
Formats disponibles : PDF | HTML | TXT

Powered by Sphinx © 2006-2012 Jean-Philippe Lemaire

Documentation du projet

Procédures

- Pdf reliés à des modes opératoires
- Documentation versionnée dans le dépôt svn



Documentation du projet

Procédures

- Pdf reliés à des modes opératoires
- Documentation versionnée dans le dépôt svn

Modes opératoires

- Accès HTML
- Reliés aux scripts et aux autres modes opératoires

Mode opératoire pour la publication d'une donnée de type raster vers un GeoServer

Jean-Baptiste Paroissien

25/09/2014

- 1 Objectif et domaine d'application
- 2 Démarche générale
- 3 Présentation des fonctions
- 4 Exemple d'application

1 Objectif et domaine d'application

Ce document présente le mode opératoire pour la publication d'un raster vers un GeoServer une fois le fichier de style sid généré. Il s'adresse aux producteurs de webservices cartographiques ayant besoin de publier un raster vers un Geoserver.

Type utilisateur	Systèmes d'exploitation	Langages
Producteurs	Linux & Windows	R et la fonction bash <code>rtogeoSERVER</code>

2 Démarche générale

La publication d'une donnée raster vers un Geoserver s'effectue à travers une suite de commandes `rtogeoSERVER` regroupées au sein de la fonction `rtogeoSERVER`. L'ensemble des arguments nécessaire pour lancer la fonction se trouve dans la table catalogue.

3 Présentation des fonctions

Les rasters sont publiés dans le Geoserver en utilisant les fonctions bash `sltogeoSERVER.sh` et `rtogeoSERVER.sh`. Les fonctions peuvent être lancées dans `R` en utilisant les arguments d'entrée de la table catalogue, voir le mode opératoire catalogue pour plus d'informations sur cette table. La table ci-dessous présente les arguments nécessaire pour l'utilisation de la fonction `rtogeoSERVER` :

Nom argument	Description
<code>replocal</code>	Répertoire de stockage des styles en local (XX/XX/)
<code>repsyle</code>	Répertoire de stockage des styles dans le serveur (XX/XX/)
<code>login_serveur</code>	Login de connexion vers le serveur (nom du serveur ou adresse IP)
<code>serveur</code>	Adresse IP du serveur
<code>nomslid</code>	Nom du fichier sid à transférer vers le serveur (sans extension)

La table ci-dessous présente les arguments d'entrée pour l'utilisation de la fonction `rtogeoSERVER` :

..

Documentation du projet

Procédures

- Pdf reliés à des modes opératoires
- Documentation versionnée dans le dépôt svn

Modes opératoires

- Accès HTML
- Reliés aux scripts et aux autres modes opératoires

Documentation script

- En-tête normalisés
- Possibilité de création d'une documentation pdf

```

@title ExportPostgis
@description Fonction permettant d'exporter les rasters tiff vers une base de données postgresql (postgres9.3-postgis2.1) + la création d'un fichier sld
@paran repertoire Chemin du répertoire de travail où se situe le paquet à créer et les fonctions à empaqueter. Format chaîne de caractères.
@paran nomraster Nom du raster à importer (ne pas oublier l'extension du fichier)
@paran paramconnexion Vecteur de configuration de connexion vers la base de données c("user", "db", "host", "port")
@paran palette Nom de la palette couleur (basée sur RColorBrewer). Pour voir les palettes : display.brewer.all(n=10, exact=FALSE)
@paran srid SRID du raster (2154 pour lambert 93)
@paran schema Schema de stockage
@paran tablerraster Nom de la table raster de stockage
@paran nomvariable Nom de la variable du raster, en général "band1"
@paran etendue Option booléenne pour la construction et l'export d'un raster étendue (TRUE) ou classique par quantile (FALSE)
@paran unite : Unité de l'élément diffusé (caractère)
@paran titre : Titre de la couche à diffuser
@paran element : Nom de l'élément diffusé
@author Jean-Baptiste Paroissien
@keywords Paquets, Documentation
@seealso \code{\link{roxygen2}}
@export
@examples
# la fonctionne pas
sourceDir(".././././Fonctions/ExporPostgis/")
nomraster <- paste(title,"grid.Gtiff",sep="")
paramconnexion <- c("postgres","delà","mayari","5435")
schema <- "data_staging"
tablerraster <- "testraster"
srid <- "2154"
nomvariable="band1" # en général, "band1"
etendue=TRUE # Sortie normal du raster, si FALSE, alors la fonction crée un raster avec les quantiles (nomraster + Gtiff)
rtpostgis(repertoire=getwd(),nomraster,srid,paramconnexion,schema,tablerraster,nomvariable,etendue=TRUE,palette="Spectral")

```

Bilan

Points positifs

- Automatisation du processus de diffusion concluant
- **Traçabilité du processus**
- Facilité à publier des données à la volée (avec des boucles)
- Documentation adaptée aux utilisateurs
 - ▶ **Liens hypertextes**
 - ▶ Mise à jour en fonction des retours

Points à revoir

- Problème sécurité, droit/gestion des accès (ssh/scp)
- Moissonnage pas encore testé
- Organisation des données diffusées dans l'IDS (geoserver + serveur)