



Gérer des métadonnées dans GeoNetwork avec la librairie R geonapi

Intervenant: Emmanuel Blondel – Consultant SI
Contact: emmanuel.blondel1@gmail.com

Atelier R métadonnées - Agropolis, Montpellier (France) 8 - 9 Février

Introduction

- Langage R
 - Large communauté d'utilisateurs: gestionnaires de données, scientifiques, statisticiens
 - Pas limité à une audience d'informaticiens et de développeurs
 - Intérêt grandissant au delà du traitement statistique:
 - automatisation de tâches de gestion de données au sens large (conversion, traitement / analyses thématiques)
 - Gestion, traitement et analyse de données spatialisées
- GeoNetwork
 - Fournit un API pour la gestion des métadonnées du catalogue
 - Deux APIs: l'ancien (stable), le nouveau (instable)
- Outils programmatiques pour la publication de métadonnées (Interfaces API)
 - Java → [geonetwork-manager](#)
 - Python → ?

Introduction

- Construire progressivement un interface R pour l'API GeoNetwork
- Références
 - <https://geonetwork-opensource.org>
 - <https://github.com/geosolutions-it/geonetwork-manager>
- Avantages
 - API fonctionnel pour la gestion de métadonnées (Création, Lecture, Mise à jour, Suppression)
 - Facile d'utilisation en combinaison avec la librairie [geometa](#)
- Inconvénients
 - Pas standard → pas lié à une norme telle que OGC CSW (Catalogue Service for the Web)
 - Donc limité à GeoNetwork! Et les autres catalogues alors?

Geonapi – le projet

- FOSS (Free and Open Source Software)
- Page web du projet: <https://github.com/eblondel/geonapi>
- Statut actuel des développements:
 - Librairie disponible uniquement disponible sur GitHub
 - Publication sur CRAN en cours d'évaluation
 - Couverture de l'API GeoNetwork: ancien API (stable)
- Perspectives de développements
 - Gestion de nouvel API
 - Création d'une librairie complémentaire qui s'appuie sur la norme CSW en vue de lever le frein lié au produit GeoNetwork → Objectif: pouvoir publier des fiches sur un serveur CSW, peu importe le "produit"
- Recherche de financements (sponsors et projets techniques)

Geonapi – Comment ça marche?

- Modèle R orienté “objet” (modèle basé sur la librairie “R6”)
- Ensemble de **classes** et de **méthodes** (“fonctions”) associées qui sont accessible via le signe “\$”.
- Principe simple d’utilisation:

1. On crée un objet d’une **classe** donnée avec la méthode **new()**:

```
monObjet <- MaClasse$new()
```

2. Puis on accède aux différentes méthodes via le “\$”. Exemple: méthode pour spécifier le nom d’un objet

```
monObjet$setName("monnom")
```

Geonapi – Chargement de la librairie

- Installer et charger la version de geonapi à partir du dépôt source Github (nécessite la librairie *devtools*)

```
library(devtools)
install_github("eblondel/geonapi")
library(geonapi)
```

Geonapi – Pratique - GNManager

- La classe **GNManager**, ou “Gestionnaire GeoNetwork” est le point d’entrée unique pour se connecter à GeoNetwork et effectuer des opérations.
- Paramètres à spécifier:
 - URL de GeoNetwork
 - Version de Geonetwork (importante car certaines opérations ont évolué avec le temps)
 - Utilisateur / Mot de passe (si nécessaire, sinon connexion anonyme)
 - Logger: NULL par défaut (aucun log), INFO: logs geonapi seulement, DEBUG: logs geonapi et CURL (logs complets des opérations)

```
GN <- GNManager$new(  
  url = "http://localhost:8080/geonetwork",  
  version = "3.0",  
  user = "admin",  
  pwd = "geonetwork",  
  logger = "DEBUG"  
)
```

Geonapi – Pratique - Publier une fiche de métadonnées

- Pré-requis: se connecter à un GeoNetwork

```
#se connecter au Geonetwork de l'atelier
GN <- GNManager$new(
  url = "http://localhost:8080/geonetwork",
  version = "3.0",
  user = "admin",
  pwd = "geonetwork",
  logger = "DEBUG"
)
```

- Objectif: Publier une fiche simple (exemple) dans le Geonetwork.
- La publication s'effectue en 2 étapes:
 - Insérer la fiche de métadonnées
 - Attribuer des privilèges à la fiche

Geonapi – Pratique - Publier une fiche de métadonnées

- Pré-requis 1: Installer la librairie uuid
 - La librairie uuid permet de créer des identifiants uniques.
 - Note: Geonetwork ne contraint pas d'utiliser ces UUIDs comme identifiant de métadonnées, mais nous l'utiliserons pour le TP afin de garantir l'unicité des identifiants dans Geonetwork

```
install.packages("uuid")
library(uuid)
uuid <- UUIDgenerate()

#on créé un nom plus explicite → remplacer "monnom"
metadata_id <- paste("eblondel", uuid, sep = "_")
```

- Pré-requis 2: Créer un début de fiche de métadonnée:

```
md = ISOMetadata$new()
md$setFileIdentifiant(metadata_id)
md$setCharacterSet("utf8")
md$setLanguage("eng")
md$setDateStamp(Sys.time())
```

Geonapi – Pratique - Publier une fiche de métadonnées

- Etape 1: Insérer la fiche
 - Fichier ou objet XML produit par geometa
 - Attribuer un groupe Geonetwork: Voir <http://localhost:8080/geonetwork/srv/eng/xml.info?type=groups>
 - Attribuer une catégorie: Voir <http://localhost:8080/geonetwork/srv/eng/xml.info?type=categories>

```
created = GN$insertMetadata (  
  xml = md$encode(),  
  group = "1",  
  category = "datasets"  
)  
created
```

```
[1] 1594
```

Identifiant interne Geonetwork

Geonapi – Pratique - Publier une fiche de métadonnées

- Etape 2: Attribuer les privilèges
 - Attribuer un groupe: "guest", "intranet" ou "all".
 - Attribuer des privilèges, parmi les valeurs suivantes:
 - "view": voir la métadonnée
 - "download": activer la partie download d'une métadonnée,
 - "editing": autoriser l'édition
 - "notify": autoriser la notification
 - "dynamic": autoriser l'affichage sur carte
 - "featured": autoriser l'affichage dans l'onglet "featured"

```
config <- GNPrivConfiguration$new()  
privileges <- c("view", "dynamic", "featured")  
config$setPrivileges(group = "all", privileges = privileges)  
GN$setPrivConfiguration(id = created, config = config)
```

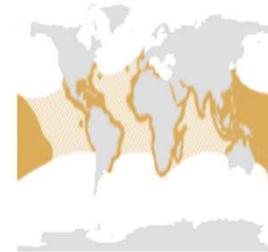
Geonapi – Pratique - Lire une fiche de métadonnées

```
#récupérer la fiche Geonetwork créée précédemment  
md_get <- GN$getMetadataByUUID(metadata_id)  
  
#et si on comparait les deux objets?  
ISOAbstractObject$compare(md, md_get)
```

Geonapi – Pratique - Lire et exploiter une fiche?



FAO AQUATIC SPECIES DISTRIBUTION MAP OF ERETMOCHELYS IMBRICATA



Abstract

The main sources of information for the species distribution are the habitat description and geographic range contained in the published FAO Catalogues of Species (more details at <http://www.fao.org/>...

Keywords

FAO, FIGIS, fishery, fisheries, aquatic species distribution, Species distribution, TTH, Hawksbill turtle, Eretmochelys imbricata, Cheloniidae, TESTUDINES, fao-species-map-tth, 3606, <http://www.fao.org/figis/lod/flod/entities/codedentity/bf7e0094cf1d2e5ae5442ec58e83352caf206bc4>

```
#connexion au FAO Geonetwork de manière anonyme
```

```
FAO = GNManager$new(url = "http://www.fao.org/geonetwork", version="2.6.4",  
"INFO")
```

```
#Récupérer la fiche de métadonnées de la distribution FAO de l'espèce de tortue  
#marine Eretmochelys imbricata (code ASFIS = TTH) et la lire avec geometa  
sp_tth <- FAO$getMetadataByUUID("fao-species-map-tth")
```

```
#Récupérer le lien OGC WFS (ESRI shapefile au format SHAPE-ZIP)
```

```
onlineResources <- sp_tth$distributionInfo$transferOptions$onLine  
shpResource <- onlineResources[sapply(onlineResources, function(x){regexpr("SHAPE-  
ZIP",x$linkage$value)>0})][[1]]  
shpLink <- shpResource$linkage$value  
download.file(shpLink, "sp_tth.zip")  
unzip("sp_tth.zip")  
library(sf)  
sp <- read_sf("SPECIES_DIST_TTH.shp")  
plot(sp, col="lightblue")
```

9 Février

Geonapi – Pratique - Mettre à jour une fiche de métadonnées

```
#récupérer la fiche Geonetwork créée précédemment
md_get <- GN$getMetadataByUUID(metadata_id)

#exemple de MAJ: ajouter l'identification des données à la fiche
ident <- ISODataIdentification$new()
ident$setAbstract("my abstract")
ident$setPurpose("purpose")
ident$setLanguage("eng")
ident$setCharacterSet("utf8")
ident$addTopicCategory("biota")
ct <- ISOCitation$new()
ct$setTitle("Ceci est mon titre à moi (monnom)")
ident$setCitation(ct)
md_get$addIdentificationInfo(ident)

#MAJ sur Geonetwork
metaId <- GN$get(md_get$fileIdentifier, by = "uuid", output = "id")
updated = GN$updateMetadata(id = metaId, xml = md_get$encode())
updated
```

Geonapi – Pratique - Supprimer une fiche de métadonnées

```
#supprimer la fiche Geonetwork créée précédemment  
deleted <- GN$deleteMetadata(metaId)  
deleted
```

Merci pour votre attention

Atelier R métadonnées - Agropolis, Montpellier (France) 8 - 9 Février