D’un GeoNature externe à ma base de données PostgreSQL : GN2PG

Des imports automatisés pour une meilleure circulation des données

Introduction

Les échanges de données permettent l’amélioration les connaissances à différentes échelles spatiales et temporelles. Ces échanges sont souvent chronophages et les outils informatiques permettent aujourd’hui de les automatiser. En France, l’outil majoritairement utilisé pour la gestion des données de biodiversité est GeoNature. Il est donc intéressant de pouvoir effectuer des liens entre une base de données GeoNature et une autre base de données (GeoNature ou autre). Ces liens peuvent être imaginés de multiples façons et la LPO Auvergne-Rhône-Alpes avait initié le développement d’une API (appelée GN2PG (GeoNature to PostGreSQL)) permettant de moissonner les données mises à disposition par le module d’export de GeoNature développé par Natural Solutions et le réseau des parcs nationaux de France.

Ce document présente cette API, testée pendant l’été 2021 dans le cadre de la mission COBIODIV SI. Il est à destination des référent.e.s SI des structures de biodiversité.

Vocabulaire

API : Interface de programmation d’application (initiales anglaises)
BDD Cible : Base de données qui récupère les données du GeoNature Source
GeoNature Cible : GeoNature (BDD Cible) qui récupère les données du GeoNature Source
GeoNature Source : GeoNature qui met à disposition ces données via le module d’export
SI : Systèmes d’informations
URL : URL (*Uniform Resource Locator)* est plus couramment appelée **adresse web**, c’est une chaîne de caractères uniforme qui permet d'identifier une ressource internet.
CRON : tâche planifiée lancée automatiquement.

Étapes de la connexion entre bases

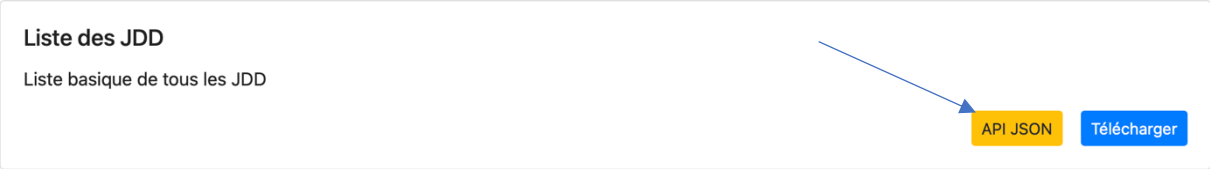
1. Mise à disposition des données par le GeoNature Source

1.1 Création d’un export – GeoNature source

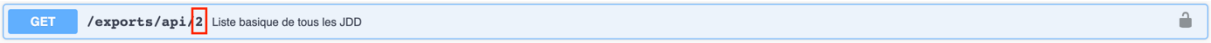
Afin de protéger les données et de sélectionner uniquement celles qui doivent être mises à disposition de la ou des structures concernée(s), la personne en charge du GeoNature source **crée un export adapté** via le module d’export :
https://github.com/PnX-SI/gn_module_export#création-dune-nouvelle-vue-dans-la-bdd

Une fois l'export créé, il est nécessaire de **récupérer l'identifiant de cet export** :

- Sur l'interface GeoNature, cliquer sur le module export



- Récupérer l'identifiant de l'export. Ici **2** par exemple.



1.2 Une fois l'identifiant récupéré, un utilisateur externe doit être créé – GeoNature source

Ceci s'effectue via l'outil UsersHub : https://github.com/PnX-SI/UsersHub
Il est possible de gérer les droits de cet utilisateur afin qu'il n'ait accès qu'au module d'export du GeoNature.
Dans le cadre de la release 2.7.5 de GeoNature, la notion d'utilisateur public peut permettre de laisser accessible des imports génériques :
https://github.com/PnX-SI/GeoNature/pull/1331/commits

1.3 Transmission des informations – GeoNature source à BDD cible

Transmettre toutes les informations à l'organisme qui importera vos données :

- URL de votre application GeoNature (exemple : https://exemple.geonature.fr/)
- ID de l'export (exemple : 2)
- Identifiant et mot de passe de l'utilisateur (exemple 'idpartenaire' et 'mdppartenaire')

2. Mise en place de l'outil GN2PG par la structure important les données

2.1 L'installation

Simplifiée, elle est précisée dans la documentation de l'outil :
https://github.com/lpoaura/GN2PG#project-setup

2.2 Après l'installation de l'API, configuration du fichier .toml (<myconfigfile>)

https://github.com/lpoaura/GN2PG#init-config-file

2.3 Description du fonctionnement

Une fois que GN2PG est installé et que le fichier .toml est configuré, il vous est demandé de lancer la commande :
gn2pg_cli --json-tables-create <myconfigfile>

Cette commande va créer un schéma (que vous avez nommé dans le fichier de configuration) dans votre base de données cible. Ce schéma contiendra plusieurs tables dont la table **data_json** qui contiendra les données importées.

La plupart des informations liées à chaque donnée (la donnée elle-même) est stockée dans la colonne **item** de la table **data_json** au format JSON.

Vous pourrez donc les stocker de cette manière où les transformer au format qui est le vôtre. (voir 2.4)

2.4 OPTIONNEL : Configuration d'un GeoNature Cible afin que les données soient intégrées dans la synthèse de ce dernier (*à lire uniquement si la base cible est un GeoNature*)

Comme énoncé dans la partie précédente, les données importées sont stockées au format JSON dans le champ **item** de la table **data_json**. Ce format est redécoupable afin de rentrer les informations dans des tables SQL. C'est l'objet du fichier **tognsynthese.sql** disponible dans le répertoire GitHub :
https://github.com/lpoaura/GN2PG/blob/c4256f4e927043e310644dbe253234d2c24678f5/gn2pg/data/to_gnsynthese.sql

Ce fichier, lorsqu'il est exécuté, permet de mettre en place des triggers dans votre base de données GeoNature cible (c'est-à-dire des fonctions qui s'exécutent à l'import de chaque de donnée). Formaté comme il l'est actuellement, il permet d'insérer automatiquement dans la synthèse les données brutes ainsi les cadres d'acquisition et les jeux de données dans le schéma **gn_meta** comme fournis par le GeoNature Source.

L'exécution de ce fichier, comme indiqué dans la documentation en en-tête, peut s'avérer difficile si des doublons sont présents dans la base du GeoNature cible. Ces doublons empêchent l'exécution de la ligne :

```
CREATE UNIQUE INDEX IF NOT EXISTS uidx_synthese_id_source_id_entity_source_pk_value
ON gn_synthese.synthese (id_source, entity_source_pk_value);
```

Cette contrainte d'unicité est nécessaire dans la mesure ou le champ "unique_id_sinp" n'est pas obligatoire dans la synthèse de GeoNature. C'est alors la paire source/id d'origine qui est privilégiée.

De fait, il est nécessaire de les supprimer pour faire tourner ce script. Dans le cadre de la mission COBIODIV SI un script de détection de données identiques a été mis en place (basé sur le modèle GeoNature et la signature d'une donnée) et permet de supprimer les doublons :
https://github.com/PnX-SI/GeoNature/pull/1341/files#diff-4b51a3a53815642b49cc52c3d8ead8d1c1cbabe62c3735b164ea5f354d5f7da8

3. Lancement de l'import

3.1 Premier import complet (https://github.com/lpoaura/GN2PG#full-download)

Une fois que les configurations sont effectuées. Un premier import peut être lancé :
gn2pg_cli --full <myconfigfile>

Cet import va regarder l'ensemble des données disponibles dans l'export programmée par le GeoNature source et les importées par 'paquets' (de 1000 par défaut, configurable dans le fichier <myconfigfile>).

3.2 Import différentiel (https://github.com/lpoaura/GN2PG#id8)

Une fois ces données importées (ce qui peut prendre un certain temps en fonction de la quantité) la commande **gn2pg_cli –update <myconfigfile>** permettra de mettre à jour la **BDD Cible** depuis le **GeoNature Source** en ne récupérant récupérant que les modifications effectuées dans le **GeoNature Source** depuis le dernier import (créations, modifications et suppressions).

Nous vous tiendrons au courant des avancées sur ces développements.

4. Automatisation de l'import

Lorsque la commande discutée en **3.2** sera officialisée, il sera possible de la lancer de manière régulière et automatique (toutes les heures, quotidiennement, hebdomadairement, mensuellement, etc.) grâce à la mise en place d'un CRON.

La mise en place de ce CRON sera décrite dans le répertoire de travail GitHub GN2PG de la LPO Aura.