

---

# GeoNature Documentation

*Release 2.0*

**PnE, pnC**

**May 19, 2022**



## CONTENTS





## INSTALLATION

GeoNature repose sur les composants suivants :

- PostgreSQL / PostGIS
- Python 3 et dépendances Python nécessaires à l'application
- Flask (framework web Python)
- Apache
- Angular 7, Angular CLI, NodeJS
- Bibliothèques javascript (Leaflet, ChartJS)
- Bibliothèques CSS (Bootstrap, Material Design)

Deux méthodes d'installation existent :

- *Installation globale* : Installation automatisée de GeoNature, TaxHub et UsersHub.
- *Installation de GeoNature uniquement* : TaxHub et UsersHub ne sont pas installés (mais leur schémas sont tous de même créés dans la base de données).

### 1.1 Prérequis

- Ressources minimum serveur :
  - Un serveur Debian 10 ou Debian 11 architecture 64-bits
  - 4 Go RAM
  - 20 Go d'espace disque
- GeoNature nécessite d'accéder à des ressources externes durant son installation et son fonctionnement. Si vous utilisez un serveur mandataire, celui-ci doit permettre l'accès aux domaines suivants :
  - <https://pypi.python.org>
  - <https://geonature.fr/>
  - <https://codeload.github.com/>
  - <https://nodejs.org/dist>
  - <https://registry.npmjs.org>
  - <https://www.npmjs.com>
  - <https://raw.githubusercontent.com/>

- <https://inpn.mnhn.fr/mtd>
- <https://preprod-inpn.mnhn.fr/mtd>
- <https://wxs.ign.fr/>

## 1.2 Préparation du serveur

Commencer la procédure en se connectant au serveur en SSH avec l'utilisateur linux `root`.

- Mettre à jour de la liste des dépôts Linux :

```
# apt update
# apt upgrade
```

- Configuration de la locale du serveur

Certains serveurs sont livrés sans “locale” (langue par défaut). Pour l’installation de GeoNature, il est nécessaire de bien configurer la locale. Si la commande `locale` renvoie ceci :

```
LANG=fr_FR.UTF-8
LANGUAGE=fr_FR.UTF-8
LC_CTYPE="fr_FR.UTF-8"
LC_NUMERIC="fr_FR.UTF-8"
LC_TIME="fr_FR.UTF-8"
LC_COLLATE="fr_FR.UTF-8"
LC_MONETARY="fr_FR.UTF-8"
LC_MESSAGES="fr_FR.UTF-8"
LC_PAPER="fr_FR.UTF-8"
LC_NAME="fr_FR.UTF-8"
LC_ADDRESS="fr_FR.UTF-8"
LC_TELEPHONE="fr_FR.UTF-8"
LC_MEASUREMENT="fr_FR.UTF-8"
LC_IDENTIFICATION="fr_FR.UTF-8"
LC_ALL=fr_FR.UTF-8
```

Vous pouvez alors passer cette étape de configuration des locales.

Sinon exécuter la commande `dpkg-reconfigure locales`. Une fenêtre s’affiche dans votre console. Dans la liste déroulante, sélectionnez `fr_FR.UTF-8 UTF-8 avec Espace`, puis cliquez sur OK. Une 2ème fenêtre s’affiche avec une liste de locale activées (`fr_FR.UTF-8` doit être présent dans la liste), confirmez votre choix, en cliquant sur OK, puis attendez que la locale s’installe.

- Installer l’utilitaire `sudo` :

```
# apt install sudo
```

- Créer un utilisateur Linux dédié (nommé `geonatureadmin` dans notre cas) pour ne pas travailler en `root` :

```
# adduser geonatureadmin
```

- Lui donner ensuite les droits administrateur en l’ajoutant au groupe `sudo` :

```
# adduser geonatureadmin sudo
```

- Pour la suite du processus d’installation, on utilisera l’utilisateur non privilégié nouvellement créé. Si besoin d’exécuter des commandes avec les droits d’administrateur, on les précèdera de `sudo`.

Il est d'ailleurs possible renforcer la sécurité du serveur en bloquant la connexion SSH au serveur avec `root`. Voir <https://docs.ovh.com/fr/vps/conseils-securisation-vps/> pour plus d'informations sur le sécurisation du serveur.

Pour passer de l'utilisateur `root` à `geonatureadmin`, vous pouvez aussi utiliser la commande :

```
# su - geonatureadmin
```

## 1.3 Installation globale

Ce document décrit une procédure d'installation packagée de GeoNature.

En lançant le script d'installation ci-dessous, l'application GeoNature ainsi que ses dépendances seront installées sur un seul et même serveur au sein d'une seule base de données.

Les applications suivantes seront installées :

- **GeoNature**
- **TaxHub** qui pilote le schéma `taxonomie`
- **UsersHub** qui pilote le schéma `utilisateurs` (le paramètre `install_usershub_app` du fichier de configuration `install_all.ini` permet de désactiver l'installation de l'application. Il est cependant recommandé d'installer l'application pour disposer d'une interface pour gérer les utilisateurs dans GeoNature)

Si vous disposez déjà de Taxhub ou de UsersHub sur un autre serveur ou une autre base de données et que vous souhaitez installer simplement GeoNature, veuillez suivre la documentation *Installation de GeoNature uniquement*.

### 1.3.1 Installation de l'application

Commencer la procédure en se connectant au serveur en SSH avec l'utilisateur dédié précédemment créé lors de l'étape de *Préparation du serveur* (usuellement `geonatureadmin`).

- Se placer à la racine du `home` de l'utilisateur puis récupérer les scripts d'installation (X.Y.Z à remplacer par le numéro de la [dernière version stable de GeoNature](#)). Ces scripts installent les applications GeoNature, TaxHub et UsersHub (en option) ainsi que leurs bases de données (uniquement les schémas du coeur) :

```
$ wget https://raw.githubusercontent.com/PnX-SI/GeoNature/X.Y.Z/install/install_all/
↪install_all.ini
$ wget https://raw.githubusercontent.com/PnX-SI/GeoNature/X.Y.Z/install/install_all/
↪install_all.sh
```

**Attention** : l'installation globale fonctionne uniquement si les scripts sont placés à la racine du `home` de l'utilisateur courant.

- Configurez votre installation en adaptant le fichier `install_all.ini` :

```
nano install_all.ini
```

Renseignez à minima votre utilisateur linux, l'URL (ou IP) de votre serveur (avec un `/` à la fin) ainsi que l'utilisateur PostgreSQL que vous souhaitez et son mot de passe. Le script se chargera d'installer PostgreSQL et de créer l'utilisateur de base de données que vous avez renseigné.

Pour la définition des numéros de version des dépendances, voir le [tableau de compatibilité](#) des versions de GeoNature avec ses dépendances. Il est déconseillé de modifier ces versions, chaque nouvelle version de GeoNature étant fournie avec les versions adaptées de ses dépendances.

- Lancer l'installation :

```
touch install_all.log
chmod +x install_all.sh
./install_all.sh 2>&1 | tee install_all.log
```

Une fois l'installation terminée, lancez la commande suivante:

```
exec bash
```

Les applications sont disponibles aux adresses suivantes :

- <http://monip.com/geonature/>
- <http://monip.com/taxhub/>
- <http://monip.com/usershub/> (en option)

Vous pouvez vous connecter avec l'utilisateur intégré par défaut (admin/admin).

**Note** Pour en savoir plus TaxHub, sa configuration et son utilisation, reportez-vous à sa documentation : <https://taxhub.readthedocs.io>. Idem pour UsersHub et sa documentation : <https://usershub.readthedocs.io>

### Note

- GeoNature-atlas compatible avec GeoNature V2 est disponible sur <https://github.com/PnX-SI/GeoNature-atlas>
- Vous pouvez utiliser le schéma `ref_geo` de GeoNature pour votre territoire, les communes et les mailles, si vous les avez intégré dans `ref_geo.l_areas` au préalable.

**Note** Une version expérimentale du calcul automatique de la sensibilité est disponible : <https://github.com/PnX-SI/GeoNature/issues/284>

Si vous rencontrez une erreur, se reporter aux fichiers de logs :

- Logs de l'installation de la base de données : `/home/`whoami`/geonature/var/log/install_db.log`
- Log général de l'installation de l'application : `/home/`whoami`/install_all.log`

Si vous souhaitez que GeoNature soit à la racine du serveur, ou à une autre adresse, éditez le fichier de configuration Apache (`/etc/apache2/sites-available/geonature.conf`) en modifiant l'alias :

- Pour `:/Alias / /home/test/geonature/frontend/dist`
- Pour `/saisie:Alias /saisie /home/test/geonature/frontend/dist`

**Note** Par défaut et par mesure de sécurité, la base de données est accessible uniquement localement par la machine où elle est installée. Pour accéder à la BDD depuis une autre machine (pour s'y connecter avec QGIS, pgAdmin ou autre), vous pouvez consulter cette documentation <https://github.com/PnX-SI/Ressources-techniques/blob/master/PostgreSQL/acces-bdd.rst>. Attention si vous redémarrez PostgreSQL (`sudo service postgresql restart`), il faut ensuite redémarrer les API de GeoNature, UsersHub et TaxHub (`sudo systemctl restart geonature.service`, `sudo systemctl restart usershub.service` et `sudo systemctl restart taxhub.service`). Attention, exposer la base de données sur internet n'est pas recommandé. Il est préférable de se connecter via un tunnel SSH. QGIS et la plupart des outils d'administration de base de données permettent d'établir une connexion à la base de cette manière.

**Note** Il est aussi important de configurer l'accès au serveur en HTTPS plutôt qu'en HTTP pour chiffrer le contenu des échanges entre le navigateur et le serveur (<https://docs.ovh.com/fr/hosting/les-certificats-ssl-sur-les-hebergements-web/>).

### 1.3.2 Installation d'un module GeoNature

L'installation de GeoNature n'est livrée qu'avec les schémas de base de données et les modules du coeur (NB : les modules Occtax, Occhab et Validation sont fournis par défaut). Pour ajouter un `gn_module` externe, il est nécessaire de l'installer :

1. Téléchargez le module depuis son dépôt Github puis dézippez-le dans le répertoire utilisateur, au même niveau que le dossier `geonature`.

```
cd /home/~whoami`
```

2. Renseignez l'éventuel fichier `config/settings.ini` du module.

3. Installez le module. Rendez-vous dans le répertoire `backend` de GeoNature et activez le `virtualenv` pour rendre disponible les commandes GeoNature :

```
source venv/bin/activate
```

Lancez ensuite la commande `geonature install_gn_module <mon_chemin_absolu_vers_le_module> <url_relative_du_module>`

Le premier paramètre est l'emplacement absolu du module sur votre serveur et le deuxième est le chemin derrière lequel on accédera au module dans le navigateur.

Exemple pour un module Import :

```
geonature install_gn_module /home/~whoami~/gn_module_import import
```

Le module sera disponible à l'adresse `http://mon-geonature.fr/geonature/#/import`

L'API du module sera disponible à l'adresse `http://mon-geonature.fr/api/import`

Cette commande exécute les actions suivantes :

- Vérification de la conformité de la structure du module (présence des fichiers et dossiers obligatoires)
- Intégration du blueprint du module dans l'API de GeoNature
- Vérification de la conformité des paramètres utilisateurs
- Génération du routing Angular pour le frontend
- Re-build du frontend pour une mise en production

4. Complétez l'éventuelle configuration du module (`config/conf_gn_module.toml`) à partir des paramètres présents dans `config/conf_gn_module.toml.example` dont vous pouvez surcoucher les valeurs par défaut. Puis relancez la mise à jour de la configuration (depuis le répertoire `geonature/backend` et une fois dans le `venv` (`source venv/bin/activate`): `geonature update_module_configuration nom_du_module`)

## 1.4 Installation de GeoNature uniquement

Cette procédure détail l'installation de GeoNature seul, sans TaxHub et UsersHub. Si vous souhaitez installer GeoNature avec TaxHub et UsersHub, reportez-vous à la section *Installation globale*.

### 1.4.1 Installation des dépendances

Installer les paquets suivants :

```
$ sudo apt install unzip git postgresql postgis python2 python3-pip python3-venv_
↳ libgdal-dev libffi-dev libpangocairo-1.0-0 apache2
```

### 1.4.2 Installation de l'application

- Se placer dans le répertoire de l'utilisateur (/home/geonatureadmin/ dans notre cas)
- Récupérer l'application (X.Y.Z à remplacer par le numéro de la [dernière version stable de GeoNature](#)). Voir le [tableau de compatibilité](#) des versions de GeoNature avec ses dépendances.

```
$ wget https://github.com/PnX-SI/GeoNature/archive/X.Y.Z.zip
```

- Dézipper l'archive de l'application

```
$ unzip X.Y.Z.zip
$ rm X.Y.Z.zip
```

- Renommer le répertoire de l'application puis placez-vous dedans :

```
$ mv GeoNature-X.Y.Z /home/`whoami`/geonature/
$ cd geonature
```

- Copier puis mettre à jour le fichier de configuration (config/settings.ini) comportant les informations relatives à votre environnement serveur :

```
$ cp config/settings.ini.sample config/settings.ini
$ nano config/settings.ini
```

### Installation de l'application

Rendez vous dans le dossier `install` et lancez successivement dans l'ordre les scripts suivant :

- `01_install_backend.sh` : Création du virtualenv python, installation des dépendances et du backend GeoNature dans celui-ci, création du service systemd (permettant d'utiliser `sudo systemctl {start, stop} geonature2`).
- `02_create_db.sh` : Création du role postgresql, de la base de données, ajout des extensions nécessaires (postgis, ...), création des schémas nécessaires à GeoNature et ajout des données métiers.
- `03_install_gn_modules.sh` : Installation des modules OccTax, OccHab et validation (si activé dans le fichier `settings.ini`).
- `04_install_frontend.sh` : Création des dossiers et liens symboliques nécessaires, création des fichier custom à partir des fichiers d'exemple, génération des fichiers de configuration grâce à la commande `geonature`, installation de nvm, npm et node ainsi que toutes les dépendances javascript nécessaires puis build du front.
- `05_configure_apache.sh` : Installation du fichier de configuration Apache /etc/apache2/conf-available/geonature.conf et activation des modules Apache nécessaires.

Vous pouvez alors démarrer le backend GeoNature : `sudo systemctl start geonature2`

## Configuration Apache

- Copiez et adaptez le fichier de configuration d'exemple d'Apache de GeoNature :

```
$ sudo cp install/assets/geonature_apache.conf /etc/apache2/sites-available/  
↳geonature.conf  
$ sudo nano /etc/apache2/sites-available/geonature.conf
```

- Activez les modules suivants :

```
$ sudo a2enmod rewrite  
$ sudo a2enmod proxy  
$ sudo a2enmod proxy_http
```

- Activez la nouvelle configuration:

```
$ sudo a2ensite geonature.conf
```

- et redémarrez Apache:

```
$ sudo systemctl restart apache2
```

- L'application est disponible à l'adresse suivante : <http://monip.com/geonature>

### 1.4.3 Dépendances

Lors de l'installation de la BDD (02\_create\_db.sh) le schéma utilisateurs de UsersHub et le schéma taxonomie de TaxHub sont intégrés automatiquement dans la BDD de GeoNature.

UsersHub n'est pas nécessaire au fonctionnement de GeoNature mais il sera utile pour avoir une interface de gestion des utilisateurs, des groupes et de leurs droits.

Par contre il est nécessaire d'installer TaxHub (<https://github.com/PnX-SI/TaxHub>) pour que GeoNature fonctionne. En effet, GeoNature utilise l'API de TaxHub. Une fois GeoNature installé, il vous faut donc installer TaxHub en le connectant à la BDD de GeoNature, vu que son schéma taxonomie a déjà été installé par le script 02\_create\_db.sh de GeoNature. Lors de l'installation de TaxHub, n'installez donc que l'application et pas la BDD.

Télécharger TaxHub depuis son dépôt Github depuis la racine de votre utilisateur :

```
cd ~  
wget https://github.com/PnX-SI/TaxHub/archive/X.Y.Z.zip  
unzip X.Y.Z.zip  
rm X.Y.Z.zip
```

en mode développeur:

```
https://github.com/PnX-SI/TaxHub.git
```

Rendez vous dans le répertoire téléchargé et dézippé, puis "désamplé" le fichier settings.ini et remplissez la configuration avec les paramètres de connexion à la BDD GeoNature précédemment installée :

```
cp settings.ini.sample settings.ini  
nano settings.ini
```

Lancer le script d'installation de l'application :

```
mkdir var
mkdir var/log
touch var/log/install_app.log
./install_app.sh 2>&1 | tee var/log/install_app.log
```

Suite à l'exécution de ce script, l'application Taxhub a été lancé automatiquement par le superviseur et est disponible à l'adresse 127.0.0.1:5000 (et l'API, à 127.0.0.1:5000/api)

Voir la doc d'installation de TaxHub : <http://taxhub.readthedocs.io/>

Voir la doc d'installation de UsersHub : <http://usershub.readthedocs.io/>

## 1.4.4 Passer en mode développement

---

**Note:** Consultez le guide *Passer en mode développement* de GeoNature.

---

## 1.5 HTTPS

La procédure décrit une méthode de certification HTTPS de votre domaine, grâce au service [Let's Encrypt](#). Les manipulations ont été effectuées sur un serveur Debian 9 avec Apache2 installé, et un utilisateur bénéficiant des droits sudo.

Ressources :

- <https://www.memoinfo.fr/tutoriels-linux/configurer-lets-encrypt-apache/>
- <https://korben.info/securiser-facilement-gratuitement-site-https.html>

### 1.5.1 Installer certbot

```
sudo apt-get install python3-certbot-apache
```

### 1.5.2 Lancer la commande cerbot

Lancer la commande suivant pour générer des certificats et des clés pour le nom de domaine que vous souhaitez mettre en HTTPS.

```
sudo certbot certonly --webroot --webroot-path /var/www/html --domain mondomaine.fr --
↪email monemail@mondomaine.fr
```

- `certonly` : demander la création du certificat uniquement.
- `--webroot` : utiliser le plugin webroot qui se contente d'ajouter des fichiers dans le dossier défini via `--webroot-path`.
- `--webroot-path` : le chemin de votre « DocumentRoot » Apache. Certbot placera ses fichiers dans `$DocumentRoot/.well-known/` pour les tests et vérifications
- `--domain` : le nom de domaine à certifier. Mettre tous les sous-domaines à certifier
- `--email` : l'adresse qui recevra les notifications de Let's Encrypt. Principalement pour rappeler de renouveler le certificat le moment venu.



### 1.5.3 Les certificats obtenus

Le certificat se trouve dans le répertoire `/etc/letsencrypt/live/mondomaine.fr/`.

Il est constitué de 4 fichiers :

- `privkey.pem` : La clé privée de votre certificat. A garder confidentielle en toutes circonstances et à ne communiquer à personne quel que soit le prétexte. Vous êtes prévenus !
- `cert.pem` : Le certificat serveur est à préciser pour les versions d'Apache < 2.4.8. Ce qui est notre cas ici.
- `chain.pem` : Les autres certificats, SAUF le certificat serveur. Par exemple les certificats intermédiaires. Là encore pour les versions d'Apache < 2.4.8.
- `fullchain.pem` : Logiquement, l'ensemble des certificats. La concaténation du `cert.pem` et du `chain.pem`. A utiliser cette fois-ci pour les versions d'Apache >= 2.4.8.

### 1.5.4 Automatiser le renouvellement du certificat

Le certificat fourni par Let's Encrypt n'est valable que 3 mois. Il faut donc mettre en place un renouvellement automatique. Ajouter une tâche automatique (Cron) pour renouveler une fois par semaine le certificat :

```
sudo crontab -e
1 8 * * Sat certbot renew --renew-hook "service apache2 reload" >> /var/log/certbot.
↪ log
```

### 1.5.5 Prise en compte des nouvelles configurations Apache

Activer les modules `ssl`, `headers` et `rewrite` puis redémarrer Apache :

```
sudo a2enmod ssl
sudo a2enmod rewrite
sudo a2enmod headers
sudo apachectl restart
```

Les fichiers de configuration des sites TaxHub et UsersHub ne sont pas à modifier, ils seront automatiquement associés à la configuration HTTPS. En revanche, la configuration de GeoNature doit être mise à jour.

### 1.5.6 Configuration de l'application GeoNature

Il est nécessaire de mettre à jour le fichier de configuration `geonature_config.toml` situé dans le répertoire `geonature/config` :

```
cd geonature/config
nano geonature_config.toml
```

Modifier les éléments suivants :

```
URL_APPLICATION = 'https://mondomaine.fr/geonature'
API_ENDPOINT = 'https://mondomaine.fr/geonature/api'
API_TAXHUB = 'https://mondomaine.fr/taxhub/api'
```

Pour que ces modifications soient prises en compte, lancer les commandes suivantes :

```
cd geonature/backend
source venv/bin/activate
geonature update_configuration
deactivate
```

Les applications sont désormais accessibles sur votre domaine sécurisé en HTTPS !

## 1.6 Taches planifiées

Depuis sa version 2.9.0, GeoNature permet de générer des profils pour chaque taxon à partir des observations existantes et validées.

Pour automatiser la mise à jour des profils en fonction des nouvelles observations, il est nécessaire de relancer automatiquement la fonction de calcul des profils de taxon en créant une tâche planifiée (cron)

Créer une tâche planifiée, exécutée tous les jours à minuit dans cet exemple :

```
sudo nano /etc/cron.d/geonature
```

Ajouter la ligne suivante en remplaçant “<CHEMIN\_ABSOLU\_VERS\_VENV>” par le chemin absolu vers le virtualenv de GeoNature et “<GEONATURE\_USER>” par l'utilisateur Linux de GeoNature :

```
0 * * * * <GEONATURE_USER> <CHEMIN_ABSOLU_VERS_VENV>/bin/geonature profiles update
```

Exemple :

```
0 * * * * geonataadmin /home/user/geonature/backend/venv/bin/geonature profiles update
```

## 1.7 Mise à jour de l'application

Attention, avant chaque mise à jour de GeoNature, il est important de sauvegarder l'application et sa base de données, ou de faire un snapshot du serveur pour pouvoir revenir à son état antérieur avant mise à jour en cas de problème.

La mise à jour de GeoNature consiste à télécharger sa nouvelle version dans un nouveau répertoire, récupérer les fichiers de configuration et de surcouche depuis la version actuelle et de relancer l'installation dans le répertoire de la nouvelle version.

La mise à jour doit être réalisée avec votre utilisateur linux courant (geonatureadmin par exemple) et non pas le super-utilisateur root.

- Télécharger la dernière version de GeoNature :

```
wget https://github.com/PnX-SI/GeoNature/archive/X.Y.Z.zip
unzip X.Y.Z.zip
rm X.Y.Z.zip
```

- Renommer l'ancien répertoire de l'application, ainsi que le nouveau :

```
mv /home/~whoami~/geonature/ /home/~whoami~/geonature_old/
mv GeoNature-X.Y.Z /home/~whoami~/geonature/
cd geonature
```

- Suivez les éventuelles notes de version spécifiques décrites au niveau de chaque version : <https://github.com/PnX-SI/GeoNature/releases>.

Si la release inclut des scripts de migration SQL : *lancer ces scripts avec l'utilisateur de BDD courant* (généralement `geonatadmin`) et non le super-utilisateur `postgres`.

Sauf mentions contraires dans les notes de version, vous pouvez sauter des versions mais en suivant bien les différentes notes de versions intermédiaires et notamment les scripts de mise à jour de la base de données à exécuter successivement.

- Si vous devez aussi mettre à jour TaxHub et/ou UsersHub, suivez leurs notes de versions mais aussi leur documentation (<https://usershub.readthedocs.io> et <https://taxhub.readthedocs.io>).
- Lancez le script de `migration.sh` à la racine du dossier `geonature`:

```
./install/migration/migration.sh
```



## MANUEL UTILISATEUR

### 2.1 Authentification

Accéder à l'application de démonstration à l'adresse <http://demo.geonature.fr/geonature>.

Connectez-vous avec l'utilisateur `admin` et le mot de passe `admin`.

### 2.2 Accueil

Vous accédez alors à la page d'accueil de l'application GeoNature.

**GeoNature - Dépôt légal de biodiversité**

Bienvenue sur GeoNature, la plateforme de saisie du dépôt légal des données de biodiversité. Elle s'adresse aux maîtres d'ouvrages concernés par le versement de données brutes de biodiversité conformément à l'article L411-1 A du Code de l'environnement (Article 7 de la loi du 8 août 2016 pour la reconquête de la biodiversité).

Cette plateforme permet de saisir les données d'occurrences de taxons alimentant les études d'évaluation préalable d'études d'évaluation préalable ou de suivi des impacts réalisées dans le cadre de l'élaboration de projet d'aménagement soumis à une évaluation environnementale donnant lieu à la production ou l'utilisation de données brutes de biodiversité.

Les données saisies dans GeoNature grâce au formulaire "Occtax" peuvent être exportées puis intégrées sur la plateforme GINCO du dépôt légal de biodiversité permettant l'agrégation, la standardisation et le versement des données d'observations de biodiversité dans l'Inventaire national du patrimoine naturel (INPN).

**La télé-procédure de versement**

Le dépôt légal de données brutes de biodiversité s'inscrit dans une démarche partenariale regroupant différents acteurs proposant plusieurs outils interconnectés. Ceux-ci assurent les tâches indispensables du processus de dépôt légal : déclarer une étude, décrire les jeux, saisir et verser les données.

- Présentation de la téléprocédure

**Plus d'informations**

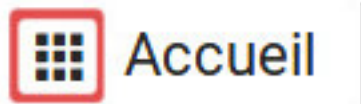
- Présentation du SINP
- INPN - Inventaire National du Patrimoine Naturel
- Projet Dépôt légal de données de biodiversité
- Documentation GeoNature

Réalisé avec GeoNature  
Développé par le Parc national des Ecrins  
et le Parc national des Cévennes

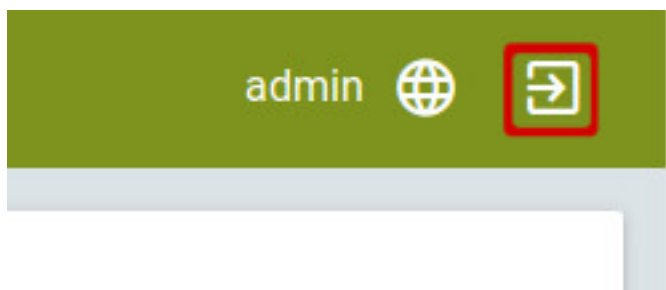
Le Menu de navigation à gauche permet d'accéder aux différents modules de l'application.



Le bouton à gauche du nom de la page permet de rabattre ou d'ouvrir le Menu de navigation.



Un bouton en haut à droite permet de se déconnecter de l'application.



## 2.3 OccTax

Ce module permet de saisir des données selon le standard Occurrence de taxon du SINP (<https://inpn.mnhn.fr/telechargement/standard-occurrence-taxon>).

Les données sont organisées en relevés (localisation, jeu de données, date, observateur...) qui sont composés d'observations d'un ou plusieurs taxons (méthode, état, statut, détermination...).

Pour chaque taxon observé, il est possible de renseigner un ou plusieurs dénombrements.

Chaque dénombrement correspond à un stade de vie et un sexe.

Chaque relevé est associé à un jeu de données. Pour saisir dans un jeu de données, il faut donc que vous ayez créé au préalable les jeux de données dans l'application Métadonnées (MTD) du SINP.

The screenshot shows the GeoNature2 web interface. On the left is a map of Europe with a blue dot indicating a location in France. On the right is a table of observations.

Taxon	Date début	Observateurs	Jeu de données
<input checked="" type="checkbox"/> Lynx lynx (Linnaeus, 1758)	17/05/2018	Administrateur test	ATBI de la réserve intégral
<input checked="" type="checkbox"/> Ablechrus caravellae Constar	18/05/2018	Administrateur test	ATBI de la réserve intégral
<input checked="" type="checkbox"/> Ablechrus caravellae Constar	18/05/2018	Administrateur test	ATBI de la réserve intégral
<input checked="" type="checkbox"/> Lynx lynx (Linnaeus, 1758), Ra	01/01/2017	Administrateur test, Adminis	Conctat aléatoire tous règ
<input checked="" type="checkbox"/> Alburnus alburnus (Linnaeus,	08/01/2017	Administrateur test	Conctat aléatoire tous règ
<input checked="" type="checkbox"/> Alburnus alburnus (Linnaeus,	17/05/2018	Administrateur test	Conctat aléatoire tous règ

6 observations au total.

Buttons: Télécharger les données, Ajouter un relevé

Quand on accède au module OccTax, celui-ci affiche vos données présentes dans le module, sur la carte ainsi que dans une liste.

La carte et la liste sont interactives.

Il est possible de se déplacer et de zoomer dans la carte (avec la souris et la molette ou les bouton + et -).

Il est aussi possible de changer le fond de carte affiché.



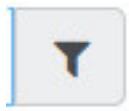
Les relevés affichés peuvent être filtrés.

Le premier filtre permet de limiter les relevés à ceux contenant un taxon en particulier.

Pour sélectionner un taxon, saisir au moins 3 lettres de l'espèce (en français ou en latin).

Il est aussi possible de saisir les premières lettres de l'espèce et de la sous-espèce.

Il est possible de filtrer la liste des taxons par règne, en affichant le filtre :



Il est possible d'afficher des filtres complémentaires, et de supprimer les filtres existants :



Cela permet de filtrer sur tous les champs du module :

Autres Critères

-- Sélectionnez un filtre --

Naturalité


Sauvage



Selon les droits dont vous disposez, il est possible d'afficher, de modifier ou de supprimer un relevé :



Sur la liste, il est également possible de modifier les colonnes affichées :

	Taxon	Date début	Observateurs	Jeu de données
<h2>Colonnes à afficher</h2> <ul style="list-style-type: none"><li><input type="checkbox"/> Altitude max</li><li><input type="checkbox"/> Altitude min</li><li><input type="checkbox"/> Commentaire</li><li><input type="checkbox"/> Date fin</li><li><input checked="" type="checkbox"/> Date début</li><li><input type="checkbox"/> ID dataset</li><li><input type="checkbox"/> ID rédacteur</li><li><input type="checkbox"/> ID relevé</li><li><input checked="" type="checkbox"/> Observateurs</li><li><input checked="" type="checkbox"/> Taxons</li></ul>				



### 2.3.1 Afficher un relevé

Si vous affichez un relevé, vous accéderez à sa fiche complète :

The screenshot displays the OccTax web interface. On the left is a map of a region in France, showing towns like Lucenay, Morance, Chazay-d'Azergues, and Belmont-d'Azergues. On the right is a panel titled 'Relevé' (Survey) with the following information:

- Relevé numéro : 4 Observateurs : Administrateur test
- Entre le : 2017-12-15 et le : 2017-12-15
- Altitude min : Altitude max :
- Taxon(s) observé(s)
  - Alburnus alburnus (Linnaeus, 1758)
  - Lynx lynx (Linnaeus, 1758)

Si vous cliquez sur un des taxons observés dans ce relevé, cela affichera le détail de l'observation du taxon :

**Taxon(s) observé(s)**

Lynx lynx (Linnaeus, 1758)	
Alburnus alburnus (Linnaeus, 1758)	

**Détails: Lynx lynx (Linnaeus, 1758)**

**Info observation** **Dénombrement**

Méthode d'observation: Vu | Etat biologique: Observé vivant

Statut biologique: Non renseigné | Naturalité: Sauvage | Méthode de détermination: Examen macroscopique terrain

Déterminateur: Gil | Statut de validation: | Preuve d'existence: Inconnu

Preuve numérique: | Preuve non numérique: |

Test

Vous pouvez aussi consulter les dénombrements du taxon observé :

**Taxon(s) observé(s)**

Lynx lynx (Linnaeus, 1758)	
Alburnus alburnus (Linnaeus, 1758)	

**Détails: Lynx lynx (Linnaeus, 1758)**

**Info observation** **Dénombrement**

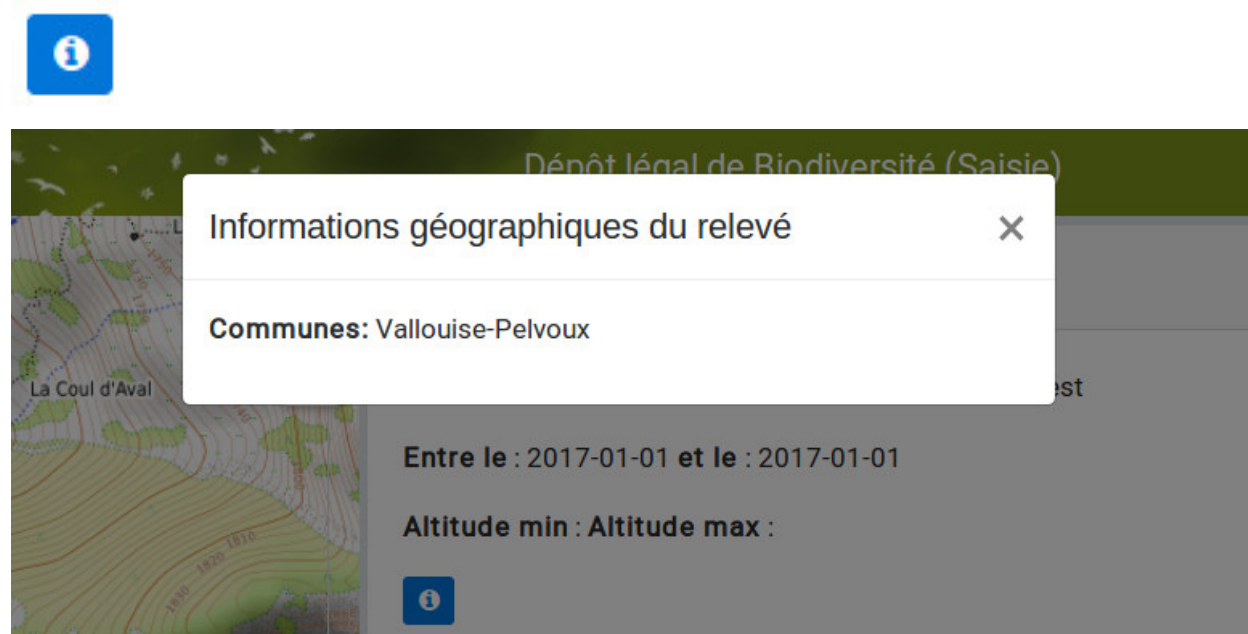
Stade de vie: Inconnu | Sexe: Non renseigné | Objet du dénombrement: Individu

Type de dénombrement: Ne sait pas | Nombre min: 5 | Nombre max: 5

Stade de vie: Inconnu | Sexe: Non renseigné | Objet du dénombrement: Individu

Type de dénombrement: Ne sait pas | Nombre min: 1 | Nombre max: 1

Il est aussi possible d'afficher les informations géographiques liées au relevé :



Selon les droits dont vous disposez, il est possible de modifier un relevé directement depuis sa fiche Détail :

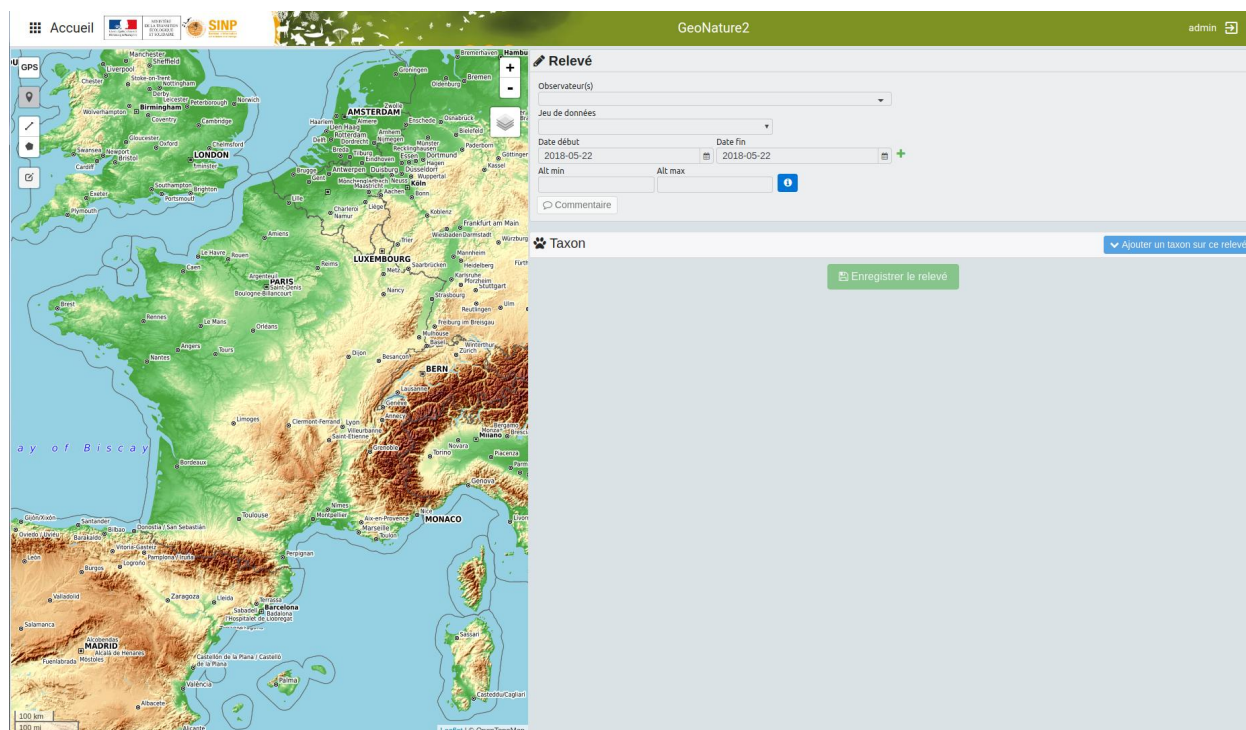


### 2.3.2 Ajouter un relevé

Depuis la liste des relevés, cliquer sur le bouton de création d'un relevé :



Vous accédez alors à un formulaire de saisie à compléter :

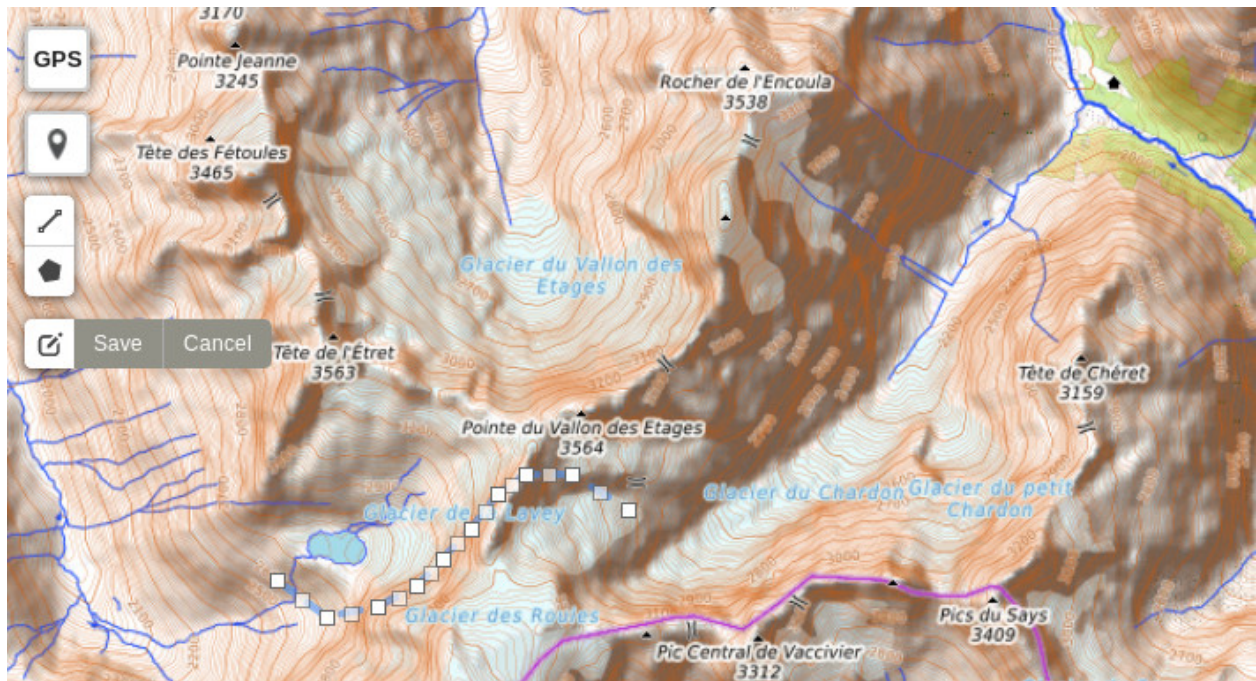


Commencez par localiser le relevé, sous forme de point (en cliquant sur la carte ou en saisissant les coordonnées GPS du point) :

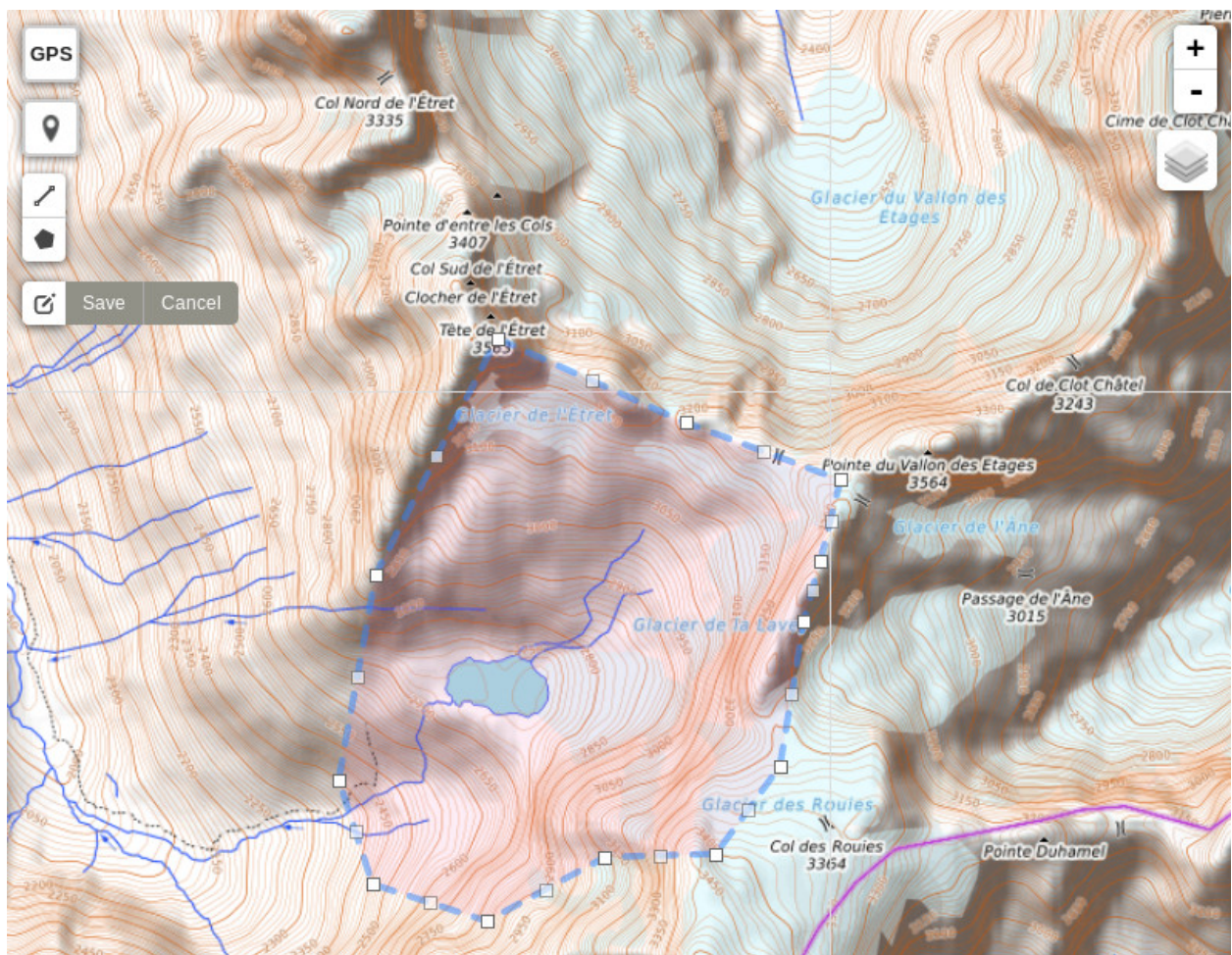


De ligne (en re cliquant sur le dernier point de la ligne pour la terminer) :





Ou de polygone (en re cliquant sur le premier point du polygone pour le terminer) :



Les localisations peuvent être modifiées.

Pour les points, il suffit de les déplacer ou de re cliquer ailleurs sur la carte.

Pour les lignes et les polygones, il faut cliquer sur le bouton de modification.



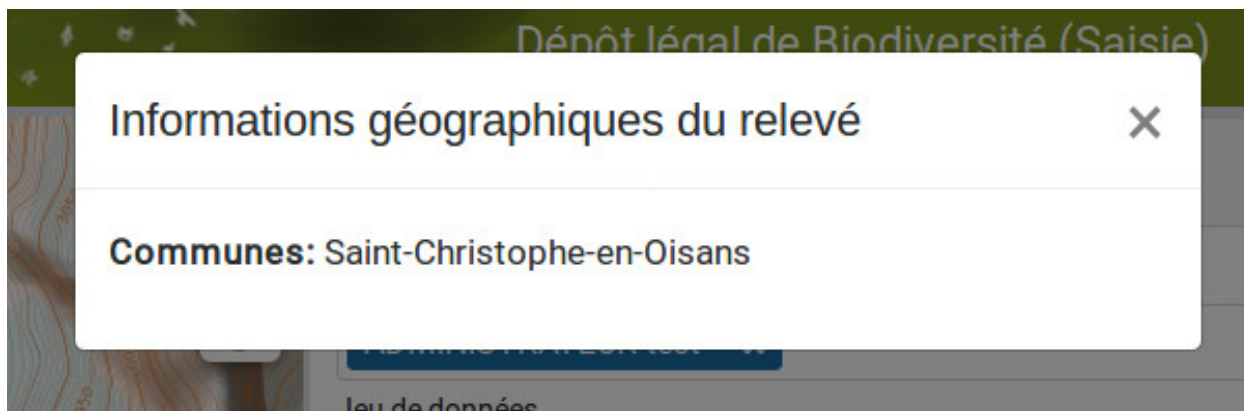
Vous pouvez alors déplacer les sommets existants ou en créer de nouveaux pour affiner le tracé en cliquant sur les sommets transparents.

Cliquer sur **SAVE** pour enregistrer les modifications apportées à une ligne ou un polygone.

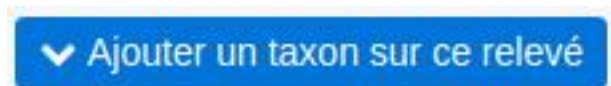
Les altitudes minimum et maximum du relevé sont calculées automatiquement mais peuvent être modifiées manuellement.

Les informations géographiques du relevé (communes notamment) sont aussi calculées automatiquement.

Pour les afficher, il faut cliquer sur le bouton d'information :



Une fois les informations du relevé renseignées (observateurs, jeu de données, date et commentaire optionnel), vous pouvez ajouter un premier taxon à celui-ci en cliquant sur **Ajouter un taxon sur ce relevé** :





The screenshot shows the GeoNature2 interface. On the left is a map with a blue location pin. On the right is a form titled 'Relevé' (Observation). The 'Taxon' section is expanded, showing a dropdown menu with 'Alburnus alburnus (Linnaeus, 1758)' selected. Below this are fields for 'Statut d'observation' (Present), 'Méthode d'observation' (Vu), and 'Etat biologique' (Observé vivant). There are also buttons for 'Avancé', 'Dénombrements', and 'Valider le taxon Alburnus alburnus (Linnaeus, 1758)'. At the bottom right is a button 'Enregistrer le relevé'.

Par défaut l'ensemble des taxons de Taxref sont disponibles à la saisie. Il est possible de restreindre cette liste pour mettre une liste personnalisée via les listes TaxHub: - au niveau du module (paramètre `id_taxon_list`. La paramètre doit être un entier correspondant à l'identifiant de la liste de la table `taxonomie.bib_listes`) - au niveau d'un jeu de données (via le formulaire de saisie des JDD, rubriques "spécificités GeoNature")

Pour sélectionner un taxon, saisissez au moins les 3 premières lettres de son nom latin ou français.

Vous pouvez aussi saisir les 3 premières lettres de l'espèce et de la sous-espèce.

Renseignez ensuite les autres champs relatifs au taxon. Les valeurs proposées dans les listes dépendent du rang et du groupe du taxon sélectionné :

This screenshot shows the 'Taxon' section of the form. It includes a dropdown menu for 'Taxon' with 'Alburnus alburnus (Linnaeus, 1758)' selected. Below are three dropdown menus: 'Statut d'observation' (Présent), 'Méthode d'observation' (Vu), and 'Etat biologique' (Observé vivant). There is a button 'Avancé' and a section for 'Dénombrements' (Counts) with fields for 'Méthode de détermination' (Autre méthode de détermination), 'Statut biologique' (Non renseigné), and 'Naturalité' (Sauvage). There are also fields for 'Détérmineur', 'Preuve d'existence' (Inconnu), 'Preuve numérique', and 'Preuve non numérique'. At the bottom is a 'Commentaire' field and a button 'Valider le taxon Alburnus alburnus'.

Des valeurs par défaut sont renseignées pour certains.

Vous pouvez ensuite renseigner un ou plusieurs dénombrements pour le taxon observé.

Par défaut, un dénombrement indéterminé d'un individu est renseigné :

**Taxon** [^ Annuler la saisie du taxon](#)

Taxon

Statut d'observation:  Méthode d'observation:  Etat biologique:

**Avancé**

Méthode de détermination:  Statut biologique:  Naturalité:

Déterminateur:

Preuve d'existence:  Preuve numérique:  Preuve non numérique:

Commentaire:






















**^ Dénombrements**

Stade de vie:  Sexe:  Objet du dénombrement:

Type de dénombrement:  Nombre min:  Nombre max:

[+ Valider le taxon Alburnus alburnus](#)

Une fois le taxon renseigné, cliquer sur **VALIDER LE TAXON** pour l'enregistrer :

Accueil                       



**Relevé**

Observateur(s): ADMINISTRATEUR test

Jeu de données: Constat aléatoire tous règnes confondus

Date début: 2018-05-22 Date fin: 2018-05-22

Alt min: Alt max:

Commentaire: Il faisait beau ...

**Taxon**

Taxon: Albumus albumus (Linnaeus, 1758)

Statut d'observation: Présent Méthode d'observation: Vu Etat biologique: Observé vivant

**Dénombrements**

Stade de vie: Adulte Sexe: Non renseigné Object du dénombrement: Individu

Type de dénombrement: Compté Nombre min: 1 Nombre max: 1

Valider le taxon Albumus albumus (Linnaeus, 1758)

Enregistrer le relevé

### 2.3.3 Modifier un relevé

Si vous modifiez un relevé existant, vous accédez à sa fiche renseignée, sur laquelle vous pouvez modifier la localisation, les informations du relevé, les taxons observés et leurs dénombrements :

**Relevé**

Observateur(s): Administrateur test

Jeu de données: Constat aléatoire tous règnes confondus

Date début: 2017-01-01 Date fin: 2017-01-01

Alt min: Alt max:

Commentaire:

**2 Taxon(s) enregistré(s)**

1	Lynx lynx (Linnaeus, 1758)	✓	✗
2	Albumus albumus (Linnaeus, 1758)	✓	✗

Enregistrer le relevé

Vous pouvez ajouter un taxon au relevé en cliquant sur Ajouter un taxon sur ce relevé :

**Relevé**

Observateur(s)  
Administrateur test

Jeu de données  
Conctat aléatoire tous règnes confondus

Date début  
2017-01-01

Date fin  
2017-01-01

Alt min

Alt max

Commentaire

**2 Taxon(s) enregistré(s)**

1	Lynx lynx (Linnaeus, 1758)		
2	Alburnus alburnus (Linnaeus, 1758)		

Taxon

Méthode d'observation  
Vu

Etat biologique  
Observé vivant

Avancé

Dénombrements

Ajouter le taxon

Enregistrer le relevé

Ou modifier une observation existante d'un taxon en le sélectionnant dans la liste des taxons déjà enregistrés :

## 2 Taxon(s) enregistré(s)

1	Lynx lynx (Linnaeus, 1758)		
2	Alburnus alburnus (Linnaeus, 1758)		

**Relevé**

Observateur(s)

1 Taxon(s) enregistré(s)  
1 Alburnus alburnus (Linnaeus, 1758)

Jeu de données  
Constat aléatoire tous règnes confondus

Date début  
2017-01-01

Date fin  
2017-01-01

Alt min

Alt max

Commentaire

**Taxon**

Taxon  
Lynx lynx (Linnaeus, 1758)

Méthode d'observation  
Vu

Etat biologique  
Observé vivant

Statut biologique  
Non renseigné

Naturalité  
Sauvage

Méthode de détermination  
Examen macroscopique terrain

Complément méthode de détermination  
Gees

Déterminateur  
Gil

Statut de validation  
En attente de validation

Preuve d'existence  
Inconnu

Preuve numérique

Preuve non numérique

Test

**^ Dénombrements**

Stade de vie  
Inconnu

Sexe  
Non renseigné

Objet du dénombrement  
Individu

Type de dénombrement  
Ne sait pas

Nombre min  
5

Nombre max  
5

Stade de vie  
Inconnu

Sexe  
Non renseigné

Objet du dénombrement  
Individu

Type de dénombrement

Nombre min

Nombre max

## 2.3.4 Exports

Une fois que vous avez saisi vos relevés et observations depuis le formulaire, vous pouvez exporter ces données en CSV selon le standard Occurrence de taxon du SINP.

Deux méthodes sont possibles pour exporter les données :

- Depuis le module “OccTax”

Depuis la liste de vos relevés de l'interface carte-liste, filtrez d'abord vos relevés par **jeu de données** (ou autre) et cliquez sur le bouton **Rechercher**.

Taxon

Observateur(s)

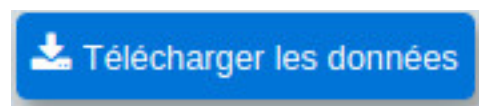
Jeu de données  
Tous

Date début

Date fin

Rechercher

Puis cliquez sur le bouton Télécharger les données en bas de la liste des relevés.



Une fenêtre s'ouvre, fournissant des informations sur le téléchargement des données. Sélectionnez ensuite le format CSV pour GINCO :



- Depuis le module d'export :

Les exports se font par jeu de données.



Vous obtenez alors un CSV par jeu de données.

Pour disposer dans l'export de toutes les informations saisies, une ligne correspond à un dénombrement d'un taxon.

Libération Sans																B I U T																A Q																%																00																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%																%															
-----------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## 2.4 Synthèse

Ce module permet de consulter, rechercher et exporter les données provenant des différentes sources et protocoles avec leur tronc commun, basé sur le standard Occurrences de taxon du SINP (<https://inpn.mnhn.fr/telechargement/standard-occurrence-taxon>).

Il permet aussi d’afficher la fiche détaillée de chaque occurrence et de revenir à sa fiche source si elle a été saisie dans un module de GeoNature.

### 2.4.1 Accéder à la synthèse

Cliquez sur le module Synthèse, dans le menu de navigation :



### 2.4.2 Présentation de la synthèse

La page principale de la synthèse est composée de 3 blocs :

- Rechercher dans les résultats
- Visualiser les résultats sur la carte
- Visualiser les résultats en liste



Rechercher Carte des résultats Liste des résultats

Synthèse

Quoi ?

Taxon

Quand ?

Date min

Date max

Où ?

Communes

Qui ?

Observateur(s)

Organisme(s)

Comment ?

Cadre d'acquisition

Jeu de données

Filtres avancés

Ajouter un filtre

Rechercher

Lancer une recherche

Supprimer tous les filtres

GeoNature2

Filtre spatial

Rechercher un lieu

Genouillet

Nis d'Aoucel

Mas Rouget

Bois de Gentile

Col de Pierre Plate

Antennes

les Baumes

le Pouchet

les Aulagnets

Valbelle

Frederic.fidon@joevannes-parcnational.fr

Taxon	Date obs	JDO	observateur
Bruant ortolan	27-07-2019	Contact aléatoire tous règnes	QUILLARD Valerie & LAMARC
Crabe à bec rouge	14-05-2019	Contact aléatoire tous règnes	DESCAMPS Régis
Bruant jaune	14-05-2019	Contact aléatoire tous règnes	DESCAMPS Régis
Athamantide de Crète	02-05-2019	Inventaire floristique vascula	MALAFOSSE Isabelle
Saxifrage des Cévennes	02-05-2019	Inventaire floristique vascula	MALAFOSSE Isabelle
Huppe fasciée	02-05-2019	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Daphné camille, Thymalée	02-05-2019	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Athamantide de Crète	02-05-2019	Inventaire floristique vascula	MALAFOSSE Isabelle
Lapin de garenne	02-05-2019	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Blanc-nez européen	02-05-2019	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Saxifrage des Cévennes	02-05-2019	Inventaire floristique vascula	MALAFOSSE Isabelle
Athamantide de Crète	02-05-2019	Inventaire floristique vascula	MALAFOSSE Isabelle
Grosbec casse-noyaux	29-01-2019	Contact aléatoire tous règnes	DESCAMPS Régis
Tarier pâtre	10-12-2018	Contact aléatoire tous règnes	FKCO Hervé
Mulet sylvestre	02-10-2018	ABC ALEPE	Gillaume BROUARD, Hugo L.
Destrier des brandes	26-07-2018	Contact aléatoire tous règnes	DESCAVES Bruno
Pie-grièche écorcheur	08-06-2018	Contact aléatoire tous règnes	HERPENA Antoine
Buisard Saint-Martin	06-06-2018	Contact aléatoire tous règnes	HERPENA Antoine
Bruant ortolan	06-06-2018	Contact aléatoire tous règnes	HERPENA Antoine
Fauvette ophée	23-05-2018	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Turcot fourmilier	23-05-2018	Contact aléatoire tous règnes	MALAFOSSE Isabelle
Buisard cendré	03-05-2018	Contact aléatoire tous règnes	DISSAC Yann & CHATELIER
Prunella verticillata	30-04-2018	ABC Florac SECHET Emman	SECHET Emmanuel, SECHET
Détail	30-04-2018	ABC Florac SECHET Emman	SECHET Emmanuel, SECHET
Lomaxe noire	07-04-2018	Contact aléatoire tous règnes	DEFFRENNES Benoit & MEYH
Milan noir	20-03-2018	Contact aléatoire tous règnes	MALAFOSSE Isabelle

0 selected / 1 596 total

Téléchargement Exporter

Par défaut, la synthèse affiche les 100 observations les plus récentes. Il est possible d'accéder aux données souhaitées en appliquant un ensemble de filtres.

### 2.4.3 Détail d'une observation

Il est possible d'accéder au détail d'une observation en cliquant sur le symbole (i) à gauche d'une observation. Le détail d'une observation correspond à l'ensemble des informations contenues dans la synthèse pour cette observation.

Une observation au sens de la synthèse ne correspond pas tout à fait à la donnée saisie initialement. C'est une représentation simplifiée et unifiée des données qui repose sur le standard SINP et qui répond aux questions suivantes :

- Où ?
- Quand ? : Date et heure de l'observation
- Qui ? : Observateur
- Quoi ? : Taxon, nombre et type d'individus, état biologique, ...
- Dans quel cadre ?

Toutes les données de la synthèse sont ramenées au niveau du dénombrement de taxon (exemple : 1 individu mâle adulte de Chevêche). Si une occurrence est constituée de 2 dénombrements, il y aura 2 enregistrements dans la synthèse (exemple : 1 individu mâle adulte et 1 individu femelle indéterminée de Chevêche).

Chouette chevêche, Chevêche d'Athéna - *Athene noctua* (Scopoli, 1769) [Voir la fiche INPN du taxon](#)

Observation de : FORT Clément  
 Date : 2019-11-15 00:00:00  
 Altitude : 1014 m - 1014 m  
 UUID observation : e13de45e-599c-47c4-af55-28ceb9f7b2aa [Voir l'observation dans le module de saisie](#)

Lien vers la fiche de saisie (si saisie sous GeoNature)

Information observation [Zonages](#) [Historique de validation](#)

Métadonnées	Détail de l'occurrence	Taxonomie
Jeu de données : Contact aléatoire tous règnes confondus	Statut de l'observation : Présent	Groupe taxonomique : Aves
Cadre : Données d'observation de la faune, de la flore et de la fonge du Parc national des Cévennes	Technique d'observation : Non renseigné	Ordre : Strigiformes
Acteurs : Parc National des Cévennes; Producteur du jeu de données	Méthode d'observation : Entendu	Famille : Strigidae
Module de provenance : Octax	Etat biologique : Observé vivant	Attribut(s) Taxonomique(s) locaux
	Statut biologique : Non renseigné	Enjeu prioritaire : oui
	Sexe : Mâle	Protégé : oui
	Stade de vie : Adulte	Réglementation
	Denombrement min : 1	Application de la Convention CITES (Convention de Washington) au sein de l'Union européenne
	Denombrement max : 1	Convention relative à la conservation de la vie sauvage et du milieu naturel de l'Europe (Convention de Berne)
	Type de dénombrement : Compté	Liste des oiseaux protégés sur l'ensemble du territoire et les modalités de leur protection
	Objet du dénombrement : Individu	
	Naturalité : Sauvage	
	Commentaire relevé	
	Commentaire occurrence	
	Déterminateur	
	Preuve d'existence : Inconnu	
	Preuve numérique	
	Preuve non numérique	
	Echantillon de preuve	
	Type de regroupement : NSP	
	Source de la donnée : Terrain	
	Statut de validation : Probable	
	Sensibilité : Maximale	
	Niveau de diffusion : Standard	
	Nature de l'objet géographique : Ne sait pas	
	Floutage : Non	

## 2.4.4 Rechercher des observations

### 1. Filtrer les données géographiquement :

Il y a 3 façons de filtrer géographiquement les données :

- en sélectionnant une commune
- en dessinant une zone sur la carte à l'aide des outils de dessin (rectangle, polygone ou cercle)
- en important un fichier de la zone

#### Filtrer par communes :

Dans le panneau filtre :

- cliquez sur le champ Communes
- saisissez les premières lettres de la commune
- sélectionnez la commune souhaitée

Il est possible de sélectionner plusieurs communes.

Ou ?

Communes

hure

Hures-la-Parade (48)

L'Estréchure (30)

Organisme(s)

Ou ?

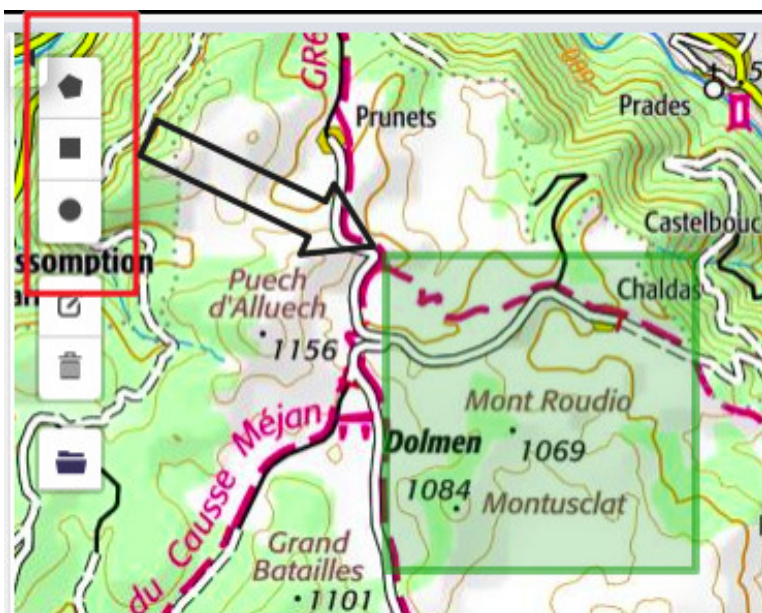
Communes

Hures-la-Parade (48)

Ou ?

#### Dessiner une zone :

Sur la carte, choisir un outil de dessin (rectangle, polygone ou cercle) et réaliser votre sélection sur la carte.



Après avoir dessiné une zone, il est nécessaire de cliquer sur le bouton Rechercher.

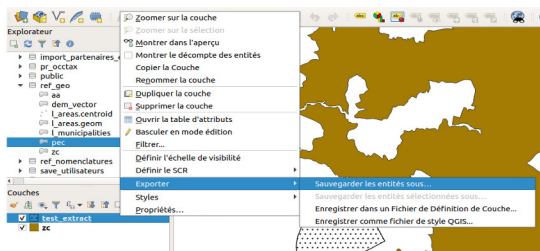
### Importer un fichier :

Il est possible d'importer une/des zone(s) de sélection directement à partir d'un fichier GeoJson.

Vous pouvez préparer ce fichier avec QGIS depuis un fichier SHP ou autre. Le fichier doit être enregistré au format GeoJson (projection 4326).

Sur la couche souhaitée :

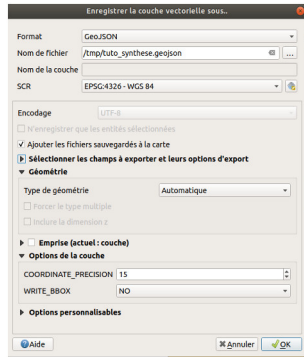
- Faire un clic droit sur la couche puis sélectionner Exporter > Sauvegarder les entités sous. . .



- Enregistrer le fichier en sélectionnant les bons paramètres :

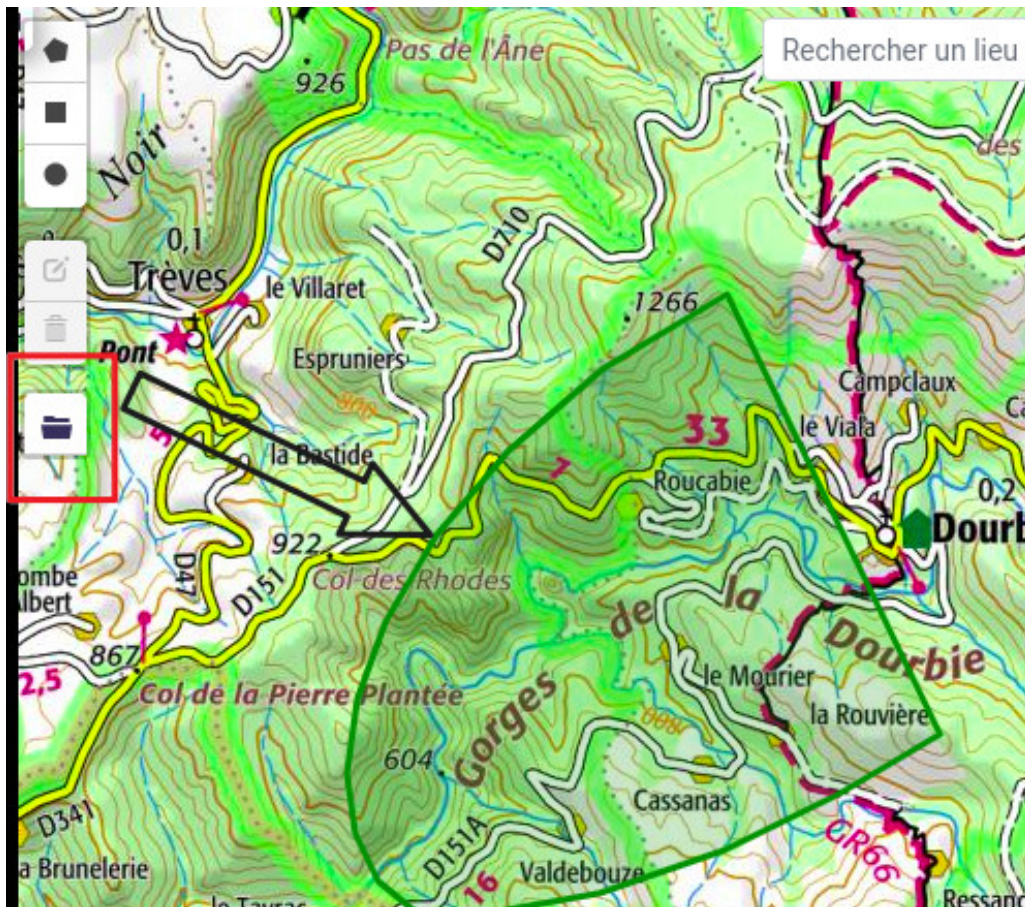
- Format : GeoJson
- SCR : WGS 84 (4326)
- Pour des questions de performance il est possible de ne pas exporter les données attributaires





Importer le fichier dans la synthèse GeoNature :

- Cliquer sur l'icône ouvrir un fichier
- Sélectionner le fichier
- La ou les zone(s) apparaissent sur la carte
- Lancer la recherche



## 2. Filtrer les données via la taxonomie

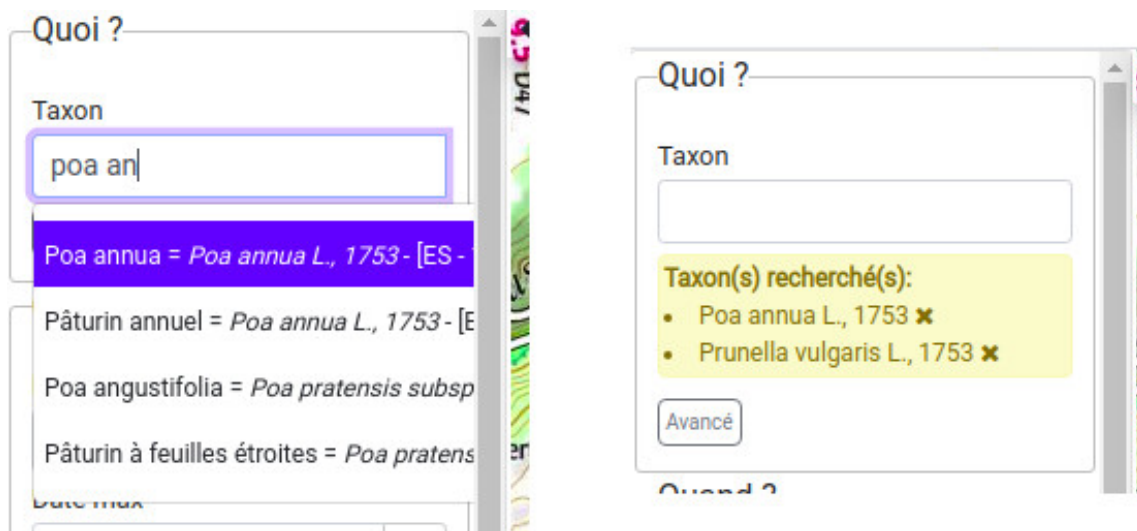
Il est possible de rechercher les données en utilisant des critères taxonomiques, en recherchant soit sur un taxon ou un groupe taxonomique en particulier, soit en se basant sur des critères taxonomiques (statut de protection, attributs)

### Recherche simple :

Dans le panneau filtre :

- cliquez sur le champ taxon
- saisissez les premières lettres du taxon
- sélectionnez le taxon souhaité

Il est possible de sélectionner plusieurs taxons.



### Recherche avancée :

Dans le panneau filtre dans la section « Quoi ? » cliquer sur Avancé.

Vous pourrez :

- Sélectionner un ou des groupes taxonomiques (exemple Chiroptera)
- Filtrer sur les listes rouge UICN,...
- Filtrer sur des attributs spécifiés dans TaxHub : patrimonialité, enjeu prioritaire, ...

Filtres taxonomiques avancés ✕

Rechercher un rang taxonomique (nom latin - au dessus du genre)

Taxon(s) recherché(s):

- Chiroptera ✕

Autre taxonomique >

Liste rouge UICN

Habitat

Groupe 2 INPN

Attributs taxhub

Evénement

Statut ZNIEFF

Statut indigène

Enjeu prioritaire

☐ oui

☐ non

Protégé

Valider et fermer

### 3. Autres filtres

Il est également possible de filtrer :

- sur une date ou une période donnée
- sur un observateur

- sur un jeu de données

The image shows a sidebar of search filters for GeoNature. On the left, four text labels with arrows point to specific filter sections:

- Filtre sur la date** points to the **Date min** and **Date max** input fields.
- Filtre sur un/des observateur(s)/organisme(s)** points to the **Observateur(s)** and **Organisme(s)** dropdown menus.
- Filtre sur les jeux de données** points to the **Cadre d'acquisition** and **Jeux de données** dropdown menus.
- Filtre sur tout autre attributs de la synthèse** points to the **ajouter un filtre** button in the **Filtres avancés** section.

The filter sidebar itself contains the following sections:

- Date min** and **Date max** (date pickers).
- Ou ?** (radio buttons for selection).
- Communes** (dropdown menu).
- Qui ?** (radio buttons for selection).
- Observateur(s)** and **Organisme(s)** (dropdown menus).
- Comment ?** (radio buttons for selection).
- Cadre d'acquisition** and **Jeux de données** (dropdown menus).
- Filtres avancés** section with an **ajouter un filtre** button.
- A red **Rechercher** button at the bottom.

## 2.5 Admin

C'est le module "backoffice" de GeoNature. Il permet notamment de gérer les permissions (CRUVED et autres filtres), les nomenclatures (typologies et vocabulaires) utilisées dans les différents modules de GeoNature ainsi que les champs additionnels.

## 2.6 Metadonnées

Ce module permet de gérer les métadonnées (Cadres d'acquisition et jeux de données), basées sur le standard Métadonnées du SINP (<http://standards-sinp.mnhn.fr/category/standards/metadonnees/>).



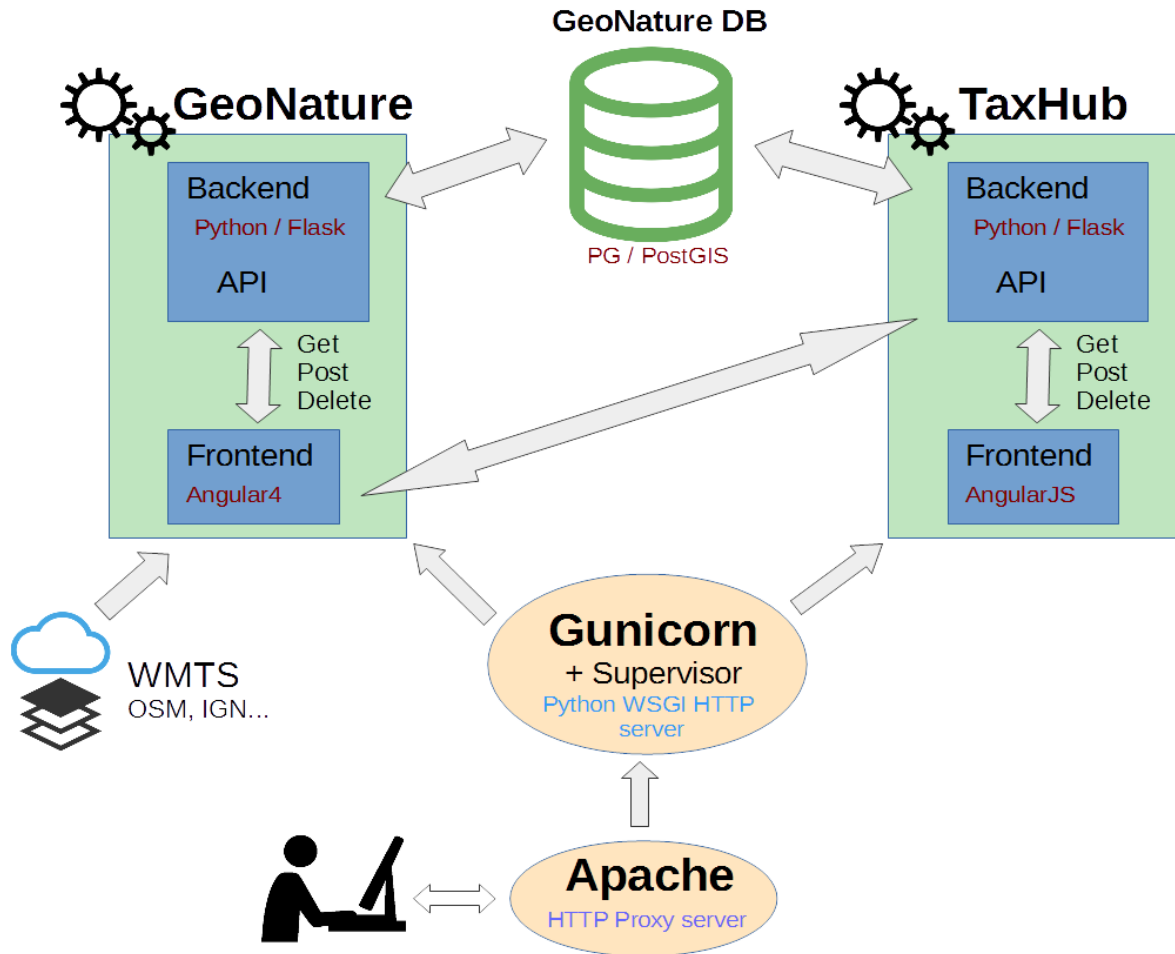
## MANUEL ADMINISTRATEUR

### 3.1 Architecture

GeoNature possède une architecture modulaire et s’appuie sur plusieurs “services” indépendants pour fonctionner :

- UsersHub et son sous-module d’authentification Flask (<https://github.com/PnX-SI/UsersHub-authentication-module>) sont utilisés pour gérer le schéma de BDD `ref_users` (actuellement nommé `utilisateurs`) et l’authentification. UsersHub permet une gestion centralisée de ses utilisateurs (listes, organismes, droits), utilisable par les différentes applications de son système d’information.
- TaxHub (<https://github.com/PnX-SI/TaxHub>) est utilisé pour la gestion du schéma de BDD `ref_taxonomy` (actuellement nommé `taxonomie`). L’API de TaxHub est utilisée pour récupérer des informations sur les espèces et la taxonomie en général.
- Un sous-module Flask (<https://github.com/PnX-SI/Nomenclature-api-module/>) a été créé pour une gestion centralisée des nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module/>), il pilote le schéma `ref_nomenclature`.
- `ref_geo` est le schéma de base de données qui gère le référentiel géographique. Il est utilisé pour gérer les zonages, les communes, le MNT, le calcul automatique d’altitude et les intersections spatiales.

GeoNature a également une séparation claire entre le backend (API: interaction avec la base de données) et le frontend (interface utilisateur). Le backend peut être considéré comme un “service” dont se sert le frontend pour récupérer ou poster des données. NB : Le backend et le frontend se lancent séparément dans GeoNature.



## 3.2 Base de données

Dans la continuité de sa version 1, GeoNature V2 utilise le SGBD PostgreSQL et sa cartouche spatiale PostGIS. Cependant l'architecture du modèle de données a été complètement revue.

La base de données a notamment été refondue pour s'appuyer au maximum sur des standards, comme le standard d'Occurrences de taxons du SINP (Voir <http://standards-sinp.mnhn.fr/category/standards/occurrences-de-taxons/>).

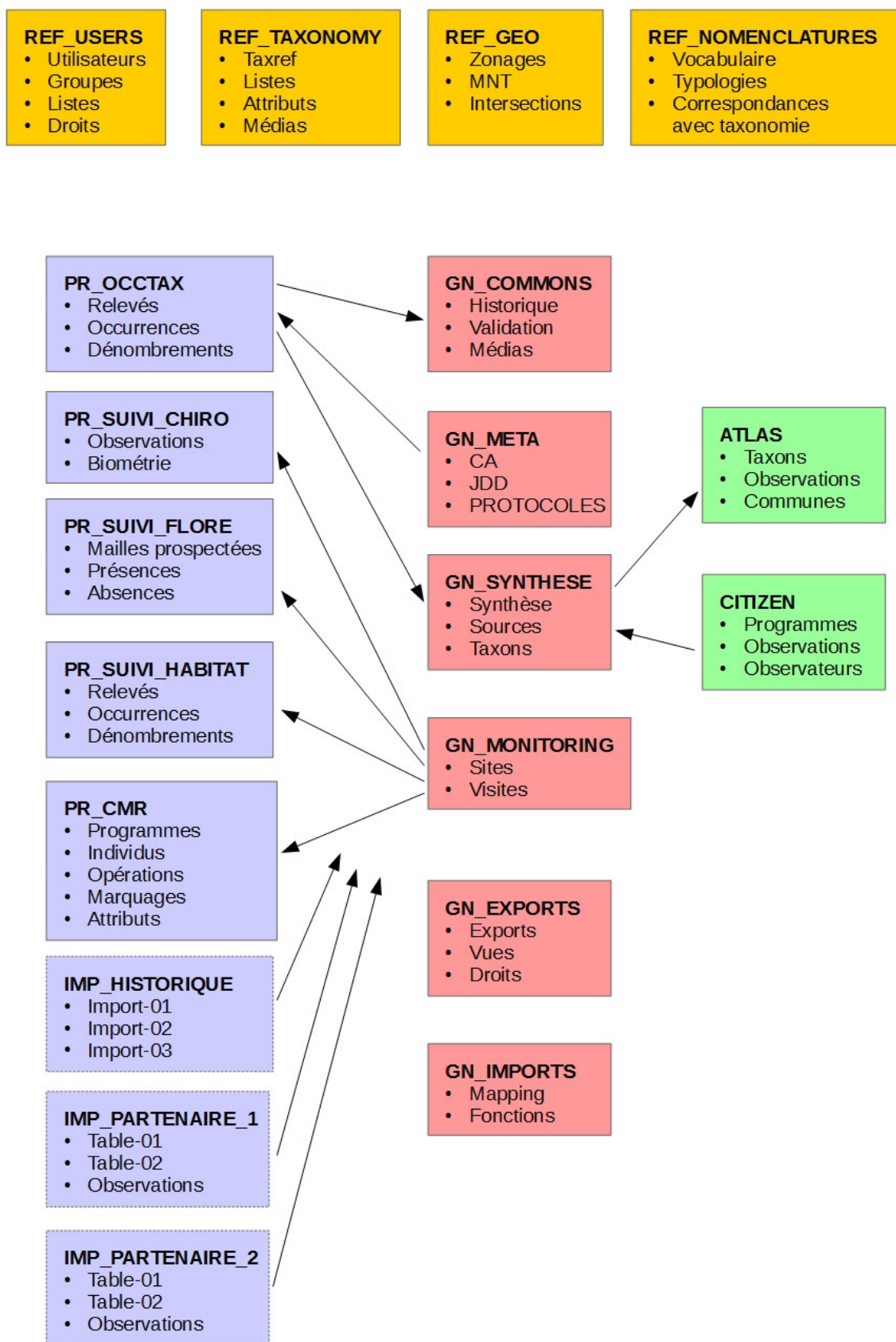
La base de données a également été traduite en Anglais et supporte désormais le multilingue.

Les préfixes des schémas de BDD sont désormais standardisés : `ref_` concerne les référentiels externes, `gn_` concerne les schémas du coeur de GeoNature et `pr_` les schémas des protocoles.

Autres standards :

- Noms de tables, commentaires et fonctions en anglais
- Pas de nom de table dans les noms de champs
- Nom de schema éventuellement dans nom de table

Schéma simplifié de la BDD :

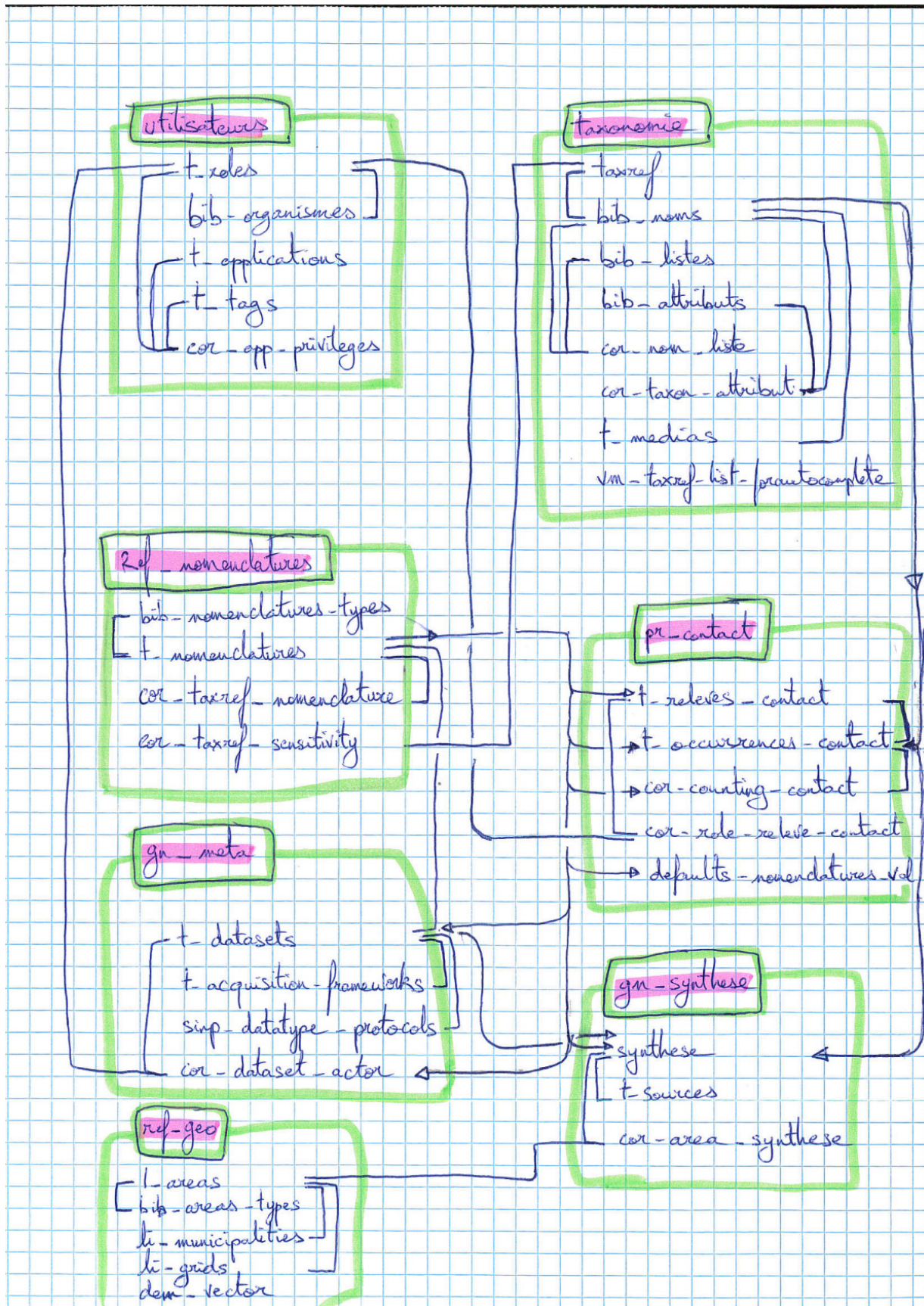


- En jaune, les schémas des référentiels.
- En rose, les schémas du coeur de GeoNature
- En bleu, les schémas des protocoles et sources de données
- En vert, les schémas des applications pouvant interagir avec le coeur de GeoNature

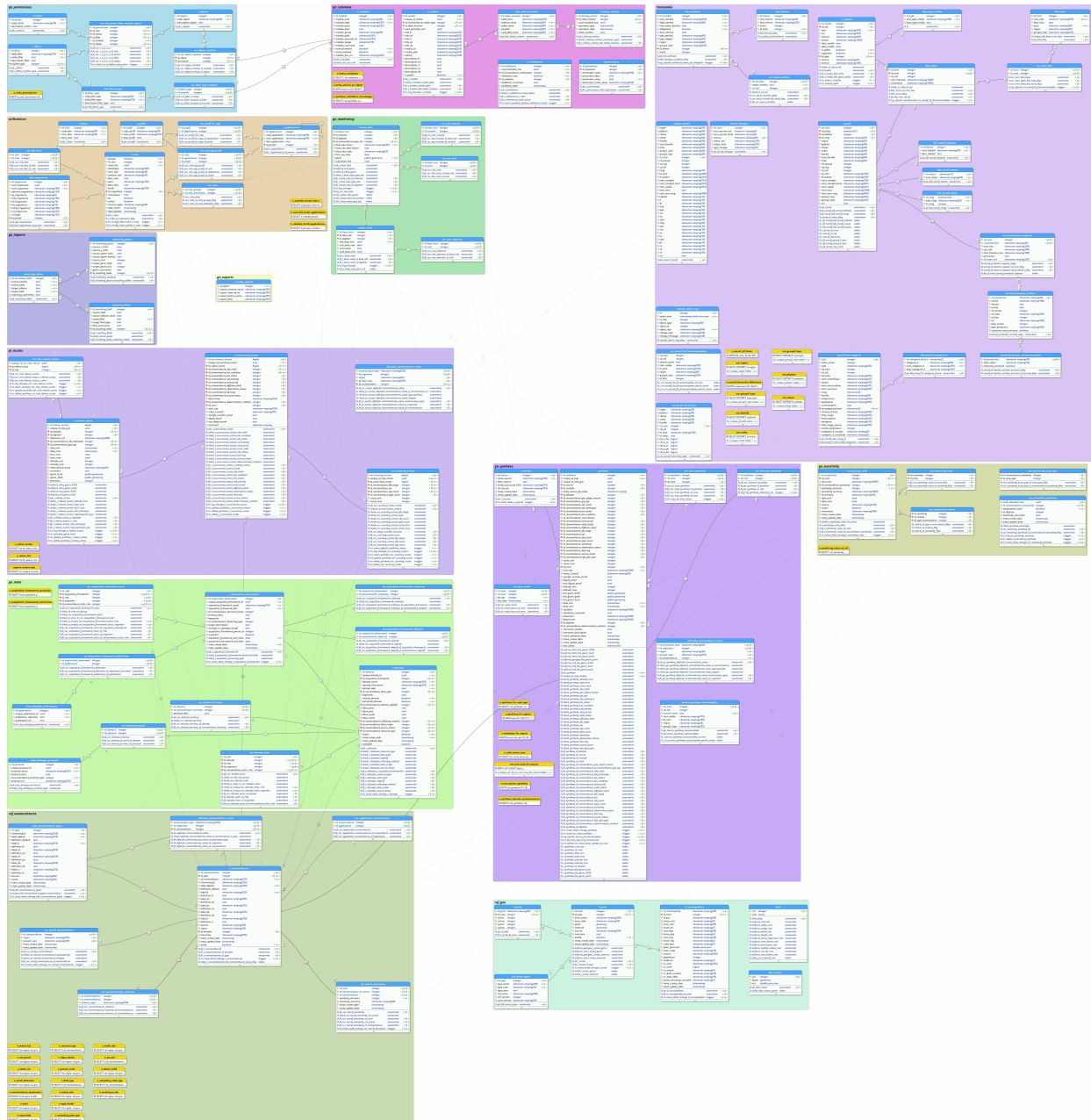
Depuis la version 2.0.0-rc.4, il faut noter que les droits (CRUVED) ont été retirés du schéma `utilisateurs` (`ref_users`) de UsersHub pour l'intégrer dans GeoNature dans un schéma `gn_permissions`, à ajouter en rose.

Modèle simplifié de la BDD (2017-12-15) :





Dernière version complète de la base de données (GeoNature 2.1 / 2019-08) :



Les relations complexes entre les schémas ont été grisées pour faciliter la lisibilité.

### 3.2.1 Administration avec Alembic

À partir de la version 2.7.5 de GeoNature, la gestion du schéma de la base de données se fait avec l'outil [Alembic](#). Celui-ci fonctionne grâce à des fichiers de migration qui sont appliqués de manière atomique (via une transaction) à la base de données, leur application étant enregistré dans la table `public.alembic_version` permettant en chaque instant de savoir dans quel état la base de données se trouve.

Les fichiers de migrations de GeoNature se trouve dans le dossier `backend/geonature/migrations/versions/`. Il est possible pour n'importe quelle dépendance ou module GeoNature de fournir également des fichiers de migrations. Pour que ceux-ci soient détectés par Alembic, il suffira de définir un point d'entrée dans le `setup.py` de la dépendance ou du module concerné :

```
setuptools.setup(
    ...,
    entry_points={
        'alembic': [
            'migrations = my_module:migrations',
        ],
    },
    ...
)
```

Il est également possible de spécifier l'emplacement de révisions Alembic manuellement dans la configuration de GeoNature. Cela est nécessaire entre autre pour UsersHub afin de pouvoir manipuler son schéma alors que UsersHub n'est usuellement pas installé dans le venv de GeoNature (seul UsersHub-authentication-module l'est) :

```
[ALEMBIC]
VERSION_LOCATIONS = '/path/to/usershub/app/migrations/versions'
```

Chaque fichier de migration est caractérisé par :

- un identifiant, *e.g.* `f06cc80cc8ba`
- une branche : Les branches permettent de séparer les fichiers de migrations afin de pouvoir les appliquer séparément. Par exemple, pour un déploiement de TaxHub sans GeoNature, il peut être intéressant de créer le schéma `taxonomie` sans créer les schémas de GeoNature, et ainsi gérer indépendamment les migrations de chaque schéma.
- un ancêtre : Lorsqu'un fichier de migration représente l'évolution d'un état antérieur de la base de données, l'ancêtre indique dans quelle version la base de données doit se trouver avant d'appliquer le-dis fichier de migration.
- des dépendances : Il est possible d'indiquer qu'une migration nécessite qu'une ou plusieurs autres migrations aient été préalablement appliquées. Par exemple, ceci permet d'indiquer que le schéma de GeoNature nécessite les schémas `taxonomie` et `utilisateurs`.

Les commandes Alembic sont disponible grâce à la sous-commande `db` de la commande `geonature` :

```
$ geonature db --help
```

Les deux sous-commandes `status` et `autoupgrade` sont spécifique à GeoNature afin d'aider à l'utilisation d'Alembic.

La commande `status` permet de visualiser les branches et l'ensemble de leurs révisions. Pour chaque révision est indiqué si celle-ci est appliqué à la base de données. Si une branche a au moins sa première révision d'appliquée, alors un petit symbole indique si cette branche est à jour, c'est-à-dire si toutes les révisions de la branche ont été appliquées (✓) ou si la branche est en retard, c'est-à-dire que celle-ci contient des révisions qui ne sont pas encore appliqué à la base de données (×).

```

[geonature ✓]
[x] f06cc80cc8ba geonature schemas 2.7.5
[x] c0fdf2ee7f4f auto update cor_area_synthese
[x] 7077aa76da3d bump dependencies
[x] 2a2e5c519fd1 fix gn_synthese.get_default_nomenclature_value
[x] 5f4c4b644844 delete cascade on cor_dataset_territory and cor_dataset_protocol
[x] 2aa558b1be3a add schema gn_profiles
[x] 1eb624249f2b add default value in additionalFields bib
[x] 7471f51011c8 change index_vm_valid_profiles_cd_ref to unique index
[x] 9a9f4971edcd fix altitude trigger
[x] 6f7d5549d49e delete view v_synthese_validation_forwebapp
[x] dde31e76ce45 remove old profile function
[x] 61e46813d621 Update synthese sensitivity
[x] dfec5f64ac73 Fix sensitivity algorithm
[x] ac08dcf3f27b Do not auto-compute diffusion_level
[x] 30edd97ae582 Remove gn_export.t_config_exports
[x] 1dbc45309d6e Merge sensitivity
[geonature-samples ✓]
[x] - 3d0bf4ee67d1 geonature samples
[habitats ✓]
[x] - 62e63cd6135d create ref_habitats schema
[habitats_inpn_data ✓]
[x] 46e91e738845 insert inpn data in ref_habitats schema
[x] 805442837a68 correction on habref data
[ign_bd_alti ✓]
[x] - 1715cf31a75d Insert default French DEM (IGN 250m BD alti)
[ign_bd_alti_vector]
[ ] - 87651375c2e8 Vectorize French DEM
[nomenclatures ✓]
[x] 6015397d686a create ref_nomenclature schema 1.3.9
[x] 11e7741319fd fix ref_nomenclatures.get_default_nomenclature_value
[x] f8c2c8482419 fix ref_nomenclatures.get_default_nomenclature_value
[x] b820c66d8daa fix ref_nomenclatures.get_nomenclature_label
[nomenclatures_inpn_data ✓]
[x] - 96a713739fdd insert inpn data in ref_nomenclatures
[nomenclatures_taxonomie ✓]
[x] - f5436084bf17 add support for taxonomy into ref_nomenclatures
[nomenclatures_taxonomie_inpn_data ✓]
[x] - a763fb554ff2 insert taxonomic inpn data in ref_nomenclatures
[occhab ✓]
[x] - 2984569d5df6 create occhab schema
[occhab-samples]
[ ] - 21f661247023 insert occhab sample data
[occtax ✓]
[x] 29c199e07eaa create occtax schema
[x] addb71d8efad create occtax export view
[x] f57107d2d0ad fix get_default_nomenclature_value
[x] 494cb2245a43 trigger comportement
[x] 944072911ff7 update synthese data (bug occtax trigger)
[occtax-samples ✓]
[x] - cce08a64eb4f insert occtax sample data
[occtax-samples-test]
[ ] - 2a0ab7644e1c occtax sample test
[ref_geo ✓]
[x] 6afe74833ed0 ref_geo schema
[x] e0ac4c9f5c0a add indexes on FK referencing l_areas.id_area
[x] 4882d6141a41 add regions in area types

```

(continues on next page)



(continued from previous page)

```

[ref_geo_fr_departments ✓]
  [x] - 3fdaa1805575 Insert French departments in ref_geo
[ref_geo_fr_municipalities ✓]
  [x] - 0dfdbfbccd63 Insert French municipalities in ref_geo
[ref_geo_fr_regions ✓]
  [x] - d02f4563bebe Insert French regions in ref_geo
[ref_geo_fr_regions_1970 ✓]
  [x] - 05a0ae652c13 Insert French regions 1970-2016 in ref_geo
[ref_geo_inpn_grids_1 ✓]
  [x] - 586613e2faeb Insert INPN 1x1 grids in ref_geo
[ref_geo_inpn_grids_10 ✓]
  [x] - ede150d9afd9 Insert INPN 10x10 grids in ref_geo
[ref_geo_inpn_grids_5 ✓]
  [x] - 7d6e98441e4c Insert INPN 5x5 grids in ref_geo
[ref_sensitivity_inpn ✓]
  [x] - 7dfd0a813f86 Insert INPN rules in sensitivity referential
[sql_utils ✓]
  [x] - 3842a6d800a0 Add public shared functions
[taxhub ✓]
  [x] - fa5a90853c45 taxhub
[taxhub-admin ✓]
  [x] - 3fe8c07741be taxhub
[taxonomie ✓]
  [x] 9c2c0254aad0 create taxonomie schema version 1.8.1
  [x] 7540702c6407 cd_ref utility functions
  [x] 98035939bc0d find_all_taxons_parents
  [x] c93cbb35cfe4 set default value for id_liste
  [x] 4fb7e197d241 create taxonomie.v_bdc_status view
[taxonomie_attributes_example]
  [ ] - aa7533601e41 add attributes exemple to taxonomie
[taxonomie_inpn_data ✓]
  [x] - f61f95136ec3 insert inpn data in taxonomie schema
[taxonomie_taxons_example]
  [ ] - 8222017dc3f6 add taxons exemple to taxonomie
[usershub ✓]
  [x] 9445a69f2bed UsersHub
  [x] 6ec215fe023e upgrade utilisateurs schema
[usershub-samples ✓]
  [x] - f63a8f44c969 UsersHub samples data
[utilisateurs ✓]
  [x] fa35dfe5ff27 utilisateurs schema 1.4.7 (usershub 2.1.3)
  [x] 830cc8f4daef add additional_data field to bib_organismes
  [x] 5b334b77f5f5 fix v_roleslist_forall_applications
  [x] 951b8270a1cf add unique constraint on bib_organismes.uuid_organisme
  [x] 10e87bcl44cd get_id_role_by_name()
[utilisateurs-samples ✓]
  [x] - 72f227e37bdf utilisateurs sample data

```

La commande `autoupgrade` permet de mettre automatiquement à jour toutes les branches dont au moins une révision est appliquée lorsque celles-ci possèdent une ou plusieurs révisions non appliquées. Cette commande est appelé par le script `migration.sh` lors d'une mise à jour de la base de données. Elle accepte également les paramètres `-x` qui sont alors fournis à la commande `upgrade` lorsque appelé sur chaque branche en retard.

La commande `heads` permet de lister l'ensemble des branches disponibles, ainsi que la dernière révision disponible pour chaque branche :

```
$ geonature db heads
e0ac4c9f5c0a (ref_geo) (effective head)
7077aa76da3d (geonature) (head)
586613e2faeb (ref_geo_inpn_grids_1) (head)
1715cf31a75d (ign_bd_alti) (effective head)
3d0bf4ee67d1 (geonature-samples) (head)
0dfdbfbccd63 (ref_geo_fr_municipalities) (head)
7d6e98441e4c (ref_geo_inpn_grids_5) (head)
87651375c2e8 (ign_bd_alti_vector) (head)
3fdaa1805575 (ref_geo_fr_departments) (effective head)
ede150d9afd9 (ref_geo_inpn_grids_10) (head)
3842a6d800a0 (sql_utils) (effective head)
951b8270alcf (utilisateurs) (effective head)
72f227e37bdf (utilisateurs-samples) (effective head)
f5436084bf17 (nomenclatures_taxonomie) (effective head)
6015397d686a (nomenclatures) (effective head)
96a713739fdd (nomenclatures_inpn_data) (effective head)
a763fb554ff2 (nomenclatures_taxonomie_inpn_data) (effective head)
4fb7e197d241 (taxonomie) (effective head)
aa7533601e41 (taxonomie_attributes_example) (head)
3fe8c07741be (taxhub-admin) (head)
8222017dc3f6 (taxonomie_taxons_example) (head)
f61f95136ec3 (taxonomie_inpn_data) (effective head)
fa5a90853c45 (taxhub) (effective head)
46e91e738845 (habitats_inpn_data) (effective head)
62e63cd6135d (habitats) (effective head)
```

La commande `history` permet de lister l'ensemble de fichier de révisions. Il est également possible de lister les révisions devant être appliqué pour passer d'un état à un autre. Par exemple, voici la liste des révisions à appliquer pour passer d'une base de données vierge (base) à une base avec la branche `nomenclatures_inpn_data` à jour (head) :

```
$ geonature db history -r base:nomenclatures_inpn_data@head
<base> (6015397d686a) -> 96a713739fdd (nomenclatures_inpn_data) (effective head),
↳ insert inpn data in ref_nomenclatures
<base> (fa35dfe5ff27, 3842a6d800a0) -> 6015397d686a (nomenclatures) (effective head),
↳ create ref_nomenclature schema 1.3.9
<base> -> 3842a6d800a0 (sql_utils) (effective head), Add public shared functions
<base> -> fa35dfe5ff27 (utilisateurs), utilisateurs schema 1.4.7 (usershub 2.1.3)
```

Si vous avez déjà une base de données existante correspondant à une installation de GeoNature en version 2.7.5 et que vous passez à Alembic, vous pouvez l'indiquer grâce à la commande suivante :

```
$ geonature db stamp f06cc80cc8ba
```

Il est possible que votre base de données contiennent quelques données supplémentaires (référentiel géographique des communes, ...), qu'il faut donc indiquer à Alembic aussi. Reportez-vous aux notes de versions de la release 2.8.0 de GeoNature afin de consulter la liste des révisions à éventuellement « *stamper* ».

Vous pouvez demander à Alembic dans quel état se trouve votre base de données avec la commande `current` :

```
$ geonature db current
62e63cd6135d (effective head)
f06cc80cc8ba
3842a6d800a0 (effective head)
9c2c0254aad
72f227e37bdf (effective head)
```

(continues on next page)

(continued from previous page)

```
fa35dfe5ff27
6015397d686a (effective head)
6afe74833ed0
a763fb554ff2 (effective head)
f5436084bf17 (effective head)
46e91e738845 (effective head)
f61f95136ec3 (effective head)
96a713739fdd (effective head)
```

La liste obtenue contient, pour chaque branche, la dernière migration appliquée. Notons toutefois que Alembic ne stocke pas l'ensemble de cette liste dans la table `public.alembic_revision`, mais se restreint uniquement aux migrations dont l'application ne peut être déduit des indications de dépendances.

Il est possible de d'afficher les informations liées à une révision avec la commande `show` :

```
$ geonature db show f06cc80cc8ba
Rev: f06cc80cc8ba
Parent: <base>
Also depends on: 72f227e37bdf, a763fb554ff2, 46e91e738845, 6afe74833ed0
Branch names: geonature
Path: backend/geonature/migrations/versions/f06cc80cc8ba_2_7_5.py

    geonature schemas 2.7.5

    Revision ID: f06cc80cc8ba
    Create Date: 2021-08-10 14:23:55.144250
```

L'absence de l'indication (head) à côté du numéro de révision indique qu'il ne s'agit pas de la dernière révision disponible pour la branche `geonature`. Vous pouvez alors mettre à jour cette branche avec la commande `upgrade` :

```
$ geonature db upgrade geonature@head
```

Il est possible de monter des branches optionnelles pour, par exemple, bénéficier des mailles 10×10 dans son référentiel géographique :

```
$ geonature db upgrade ref_geo_inpn_grids_10@head -x data-directory=./tmp_geo
```

L'ensemble des branches disponible est décrit dans la sous-section ci-après.

L'argument `-x` permet de fournir des variables à usage des fichiers de migrations. Dans le cas des migrations de données de zones géographiques, celles-ci supporte la variable `data-directory` permettant de spécifier où doivent être cherché et éventuellement téléchargé les données géographiques. Si l'argument n'est pas spécifié, un dossier temporaire, supprimé à la fin de la procédure, sera utilisé.

Pour supprimer les mailles 10×10 de son référentiel géographique, on utilisera :

```
$ geonature db downgrade ref_geo_inpn_grids_10@base
```

Dans le cas d'une branche contenant plusieurs migrations, on pourra appliquer ou dé-appliquer chaque migration individuellement avec `upgrade branch@+1` ou `downgrade branch@-1`. Il est également possible de référencer directement un numéro de migration.

Si l'on souhaite appliquer une migration manuellement, ou si l'on souhaite la modifier, il est possible de passer l'argument `--sql` aux commandes `upgrade` et `downgrade` afin de récupérer le code SQL de la migration. Cela ne fonctionne toutefois pas avec certaines migrations telles que les migrations de données géographique en raison d'import SQL nécessitant de manipuler directement le curseur `SQLAlchemy`.

Pour créer un nouveau fichier de migration afin d'y placer ses évolutions de la base de données, on utilisera la commande suivante :

```
$ geonature db revision -m "add table gn_commons.t_foo" --head geonature@head
Generating [...]/backend/geonature/migrations/versions/31250092bce3_add_table_gn_
↪commons_t_foo.py ... done
```

La [documentation d'Alembic](#) liste les opérations prises en charge. Certaines opérations complexes telles que la création de trigger ne sont pas prévu, mais il reste toujours possible d'exécuter du SQL directement avec l'opérateur `op.execute`.

### Description des branches

Cette section liste les branches Alembic disponibles et leur impacte sur la base de données.

- `sql_utils` : Fournie quelques fonctions SQL utilitaires dans le schéma `public`. Fournie par `Utils-Flask-SQLAlchemy`.
- `geonature` : Crée les schémas propres à GeoNature (`gn_commons`, `gn_synthese`, ...).
- `geonature-samples` : Insert quelques données d'exemple en base.
- `taxonomie` : Crée le schéma `taxonomie`. Fournie par `TaxHub`.
- `taxonomie_inpn_data` : Insert le référentiel `TAXHUBv14` en base. Fournie par `TaxHub`.
- `taxonomie_attributes_example` : Insert quelques attributs d'exemple en base. Fournie par `TaxHub`.
- `taxonomie_taxons_example` : Insert quelques taxons d'exemple en base. Fournie par `TaxHub`.
- `nomenclatures` : Crée le schéma `ref_nomenclatures`. Fournie par `Nomenclature-api-module`.
- `nomenclatures_inpn_data` : Insert le référentiel des nomenclatures de l'INPN en base. Fournie par `Nomenclature-api-module`.
- `nomenclatures_taxonomie` : Complète le schéma `ref_nomenclatures` pour accueillir les nomenclatures liées à la taxonomie.
- `nomenclatures_taxonomie_inpn_data` : Insert les nomenclatures liées à la taxonomie en base.
- `utilisateurs` : Installe le schéma `utilisateurs`. Fournie par `UsersHub-authentification-module`.
- `utilisateurs-samples` : Insert des données d'exemples (`utilisateurs`, `groupes`) dans le schéma `utilisateurs`. Fournie par `UsersHub-authentification-module`.
- `habitats` : Crée le schéma `ref_habitats`. Fournie par `Habref-api-module`.
- `habitats_inpn_data` : Insert le référentiel `HABREF` de l'INPN en base. Fournie par `Habref-api-module`.
- `ref_geo` : Crée le schéma `ref_geo`.
- `ref_sensitivity_inpn` : Insère le référentiel de sensibilité de l'INPN en base.

Si vous utilisez `TaxHub`, vous pouvez être intéressé par les branches suivantes :

- `taxhub` : Déclare l'application `TaxHub` dans la liste des applications. Fournie par `TaxHub`.
- `taxhub-admin` : Associe le groupe « `Grp_admin` » issue des données d'exemple à l'application `UsersHub` et au profil « `Administrateur` » permettant aux utilisateurs du groupe de se connecter à `TaxHub`. Fournie par `TaxHub`.

Si vous utilisez `UsersHub`, vous pouvez être intéressé par les branches suivantes :

- `usershub` : Déclare l'application `UsersHub` dans la liste des applications. Fournie par `UsersHub`.



- `usershub-samples` : Associe le groupe « Grp\_admin » issue des données d'exemple à l'application UsersHub et au profil « Administrateur » permettant aux utilisateurs du groupe de se connecter à UsersHub. Fournie par UsersHub.

Les branches ci-après sont totalement optionnelles :

- `ref_geo_inpn_grids_1` : Insert les mailles 1×1 km (INPN) dans le référentiel géographique (type de zone M1).
- `ref_geo_inpn_grids_5` : Insert les mailles 5×5 km (INPN) dans le référentiel géographique (type de zone M5).
- `ref_geo_inpn_grids_10` : Insert les mailles 10×10 km (INPN) dans le référentiel géographique (type de zone M10).
- `ref_geo_fr_municipalities` : Insert les municipalités française (IGN février 2020) dans le référentiel géographique (type de zone COM).
- `ref_geo_fr_departments` : Insert les départements français (IGN février 2020) dans le référentiel géographique (type de zone DEP).
- `ign_bd_alti` : Insert le modèle numérique de terrain (MNT) de l'IGN en base.
- `ign_bd_alti_vector` : Vectorise le MNT.

Note : pour plusieurs fichiers de révisions, notamment lié au référentiel géographique ou nécessitant des données INPN, il est nécessaire de télécharger des ressources externes. Il est possible d'enregistrer les données téléchargé (et ne pas les re-télécharger si elles sont déjà présentes) avec `-x data-directory=...` :

```
$ geonature db upgrade ...@head -x data-directory=./data/
```

### 3.2.2 Gestion des droits

#### Accès à GeoNature et CRUVED

Les comptes des utilisateurs, leur mot de passe, email, groupes et leur accès à l'application GeoNature est géré de manière centralisée dans UsersHub. Pour qu'un rôle (utilisateur ou groupe) ait accès à GeoNature, il faut lui attribuer un profil de "Lecteur" dans l'application GeoNature, grâce à l'application UsersHub.

La gestion des droits (permissions) des rôles, spécifique à GeoNature, est ensuite gérée dans un schéma (`gn_permissions`) et un module de GeoNature dédié.

Les permissions des groupes et utilisateurs peuvent en effet être administrées dans le module "Admin / Administration des permissions" de GeoNature. Dans la version 1 de GeoNature, il était possible d'attribuer des droits selon 6 niveaux à des rôles (utilisateurs ou groupes). Pour la version 2 de GeoNature, des évolutions ont été réalisées pour étendre les possibilités d'attribution de droits et les rendre plus génériques.

La gestion des droits dans GeoNature, comme dans beaucoup d'applications, est liée à des actions (Create / Read / Update / Delete aka CRUD). Pour les besoins métiers de l'application nous avons rajouté deux actions : "Valider" et "Exporter", ce qui donne le CRUVED : Create / Read / Update / Validate / Export / Delete.

Sur ces actions, on peut appliquer des "portées":

- **Portée 1 = "Mes données". Cela concerne les données sur lesquels je suis :**
  - observateur
  - personne ayant effectuée la saisie de la données
  - personnellement acteur du jeu de données de la donnée
  - personne ayant saisi le JDD de la donnée

- **Portée 2 = Les données de mon organisme. Portée 1 + :**
  - les données sur lesquels mon organisme est acteur du JDD de la donnée
- **Portée 3 = Toute les données**
  - Toute les données : aucun filtre n'est appliqué

Exemple :

- Utilisateur 1 peut effectuer l'action "DELETE" sur la portée "SES DONNEES"
- Utilisateur Admin peut effectuer l'action "UPDATE" sur la portée "TOUTES LES DONNEES"

Enfin ces permissions vont pouvoir s'attribuer à l'ensemble de l'application GeoNature et/ou à un module.

On a donc le quadriptyque : Un utilisateur / Une action / Une portée / Un module

**NB :** certains objets comme les JDD et CA sont transversal à tout GeoNature (ils sont utilisés dans tous les modules: saisie, synthese, métadonnées, dashbord), il sont donc contrôlé par les permissions du "module" GeoNature

### Cas particulier de l'action "C"

Dans les modules de saisie, on veut que des utilisateurs puissent saisir uniquement dans certains JDD.

La liste des JDD ouvert à la saisie est contrôlée par l'action "CREATE" du module dans lequel on se trouve.

Comme il n'est pas "normal" de pouvoir saisir dans des JDD sur lesquels on n'a pas le droit de lecture, la portée de l'action "CREATE" vient simplement réduire la liste des JDD sur lesquels on a les droits de lecture ("READ").

Même si la portée de l'action "CREATE" sur le module est supérieure à l'action "READ", l'utilisateur ne verra que les JDD sur lesquels il a des droits de lecture

### Récapitulatif

- Dans GeoNature V2 on peut attribuer à un rôle des actions possibles, sur lesquels on peut ajouter des filtres, dans un module ou sur toute l'application GeoNature (définis dans `gn_permissions.cor_role_action_filter_module_object`).
- 6 actions sont possibles dans GeoNature : Create / Read / Update / Validate / Export / Delete (aka CRUVED).
- Différents types de filtre existent. Le plus courant est le filtre de type "SCOPE" (portée) : 3 portées sont attribuables à des actions: Mes données / Les données de mon organisme / Toutes les données.
- Une vue permet de retourner toutes les actions, leurs filtres et leurs modules de GeoNature pour tous les rôles (`gn_permissions.v_users_permissions`)
- Des fonctions PostgreSQL ont aussi été intégrées pour faciliter la récupération de ces informations (`gn_permissions.cruved_for_user_in_module`, `gn_permissions.does_user_have_scope_permission`, ...)
- Les permissions attribuées à un module surchargent les permissions attribuées sur l'ensemble de l'application par un mécanisme d'héritage. Par défaut et en l'absence de permissions, tous les modules héritent des permissions de GeoNature. Attention cependant aux utilisateurs appartenant à plusieurs groupes. Si un CRUVED est défini pour un module à un seul de ses groupes, c'est ce CRUVED qui sera pris en compte. En effet, le mécanisme d'héritage ne fonctionne plus lorsqu'on surcouché implicitement le CRUVED d'un module pour un groupe.
- Si un utilisateur n'a aucune action possible sur un module, alors il ne lui sera pas affiché et il ne pourra pas y accéder
- Il est aussi possible de ne pas utiliser UsersHub pour gérer les utilisateurs et de connecter GeoNature à un CAS (voir configuration). Actuellement ce paramétrage est fonctionnel en se connectant au CAS de l'INPN (MNHN)



### La gestion des droits par le CRUVED

Actions	<b>C</b> reate	<b>R</b> ead	<b>U</b> pdate	<b>V</b> alidate	<b>E</b> xport	<b>D</b> eleter
Portée	<b>0</b> Aucune	<b>1</b> Mes données	<b>2</b> Données de mon organisme	<b>3</b> Toutes les données		

#### S'attribut:

- à un utilisateur ou à un groupe
- au niveau de GeoNature et d'un module

	GeoNature	Module occtax
Personnels en poste	<b>C: 3 R: 3 U: 1 V:1 E: 2 D: 1</b>	-
Partenaire X	<b>C: 2 R: 2 U: 1 V:1 E: 2 D: 1</b>	<b>C: 0 R: 0 U: 0 V:0 E: 0 D: 0</b>

A noter que toutes les actions et toutes les portées n'ont pas été implémentées dans tous les modules. Elles le sont en fonction des besoins de chaque module.

TODO : Lister les permissions implémentées dans chaque module.

### 3.2.3 Accès public

Cette section de la documentation concerne l'implémentation d'un utilisateur-lecteur pour votre instance GeoNature, permettant d'y donner accès sans authentification.

Étapes :

1/ UsersHub :

- Aller dans la section *Utilisateurs*
- Créer un utilisateur
- Définir un identifiant et un mot de passe (par défaut utilisateur 'public' et mot de passe 'public')
- Aller ensuite dans la section *Applications*
- Pour GeoNature, cliquer sur le premier icône 'Voir les membres'
- Cliquer sur ajouter un rôle
- Choisir l'utilisateur juste créé
- Attribuer le rôle 1, 'lecteur'

2/ Configuration GeoNature :

- Reporter identifiant et mot de passe dans le fichier de configuration de GeoNature (`config/geonature_config.toml`)

```
PUBLIC_LOGIN = 'public'
PUBLIC_PASSWORD = 'public'
```

- Mettre à jour la configuration de GeoNature

```
$ source backend/venv/bin/activate
$ geonature update_configuration
$ sudo systemctl restart geonature
```

A ce moment-là, cet utilisateur a tous les droits sur GeoNature. Il s'agit maintenant de gérer ses permissions dans GeoNature même.

### 3/ GeoNature

- Se connecter à GeoNature avec un utilisateur administrateur
- Aller dans le module Admin
- Cliquer sur 'Gestion des permissions'
- Choisissez l'utilisateur sélectionné
- Editer le CRUVED pour chacun des modules de l'instance. Passer à 0 tous les droits et tous les modules devant être supprimés. Laisser '3' pour les modules d'intérêt.

## 3.2.4 Nomenclatures

- Toutes les valeurs des listes déroulantes sont gérées dans une table générique `ref_nomenclatures`. `t_nomenclatures`
- Elles s'appuient sur les nomenclatures du SINP (<http://standards-sinp.mnhn.fr/nomenclature/>) qui peuvent être désactivées ou complétées
- Chaque nomenclature est associée à un type, et une vue par type de nomenclature a été ajoutée pour simplifier leur usage
- Ces nomenclatures sont gérées dans un sous-module pour pouvoir les réutiliser (ainsi que leur mécanisme) dans d'autres applications : <https://github.com/PnX-SI/Nomenclature-api-module/>
- Les identifiants des nomenclatures et des types de nomenclature sont des serials (entiers auto-incrémentés) et ne sont pas prédéfinis lors de l'installation, ni utilisées en dur dans le code des applications. En effet, les nomenclatures peuvent varier en fonction des structures. On utilise le `cd_nomenclature` et le mnémonique du type de nomenclature pour retrouver dynamiquement l'`id_nomenclature` d'une nomenclature. C'est cependant cet identifiant qu'on stocke au niveau des données pour garantir l'intégrité référentielle
- Chaque nomenclature peut être associée à un règne ou un `group2inpn` (`ref_nomenclatures.cor_taxref_nomenclature`) pour proposer des nomenclatures correspondants à un taxon
- Les valeurs par défaut sont définies dans chaque module
- Pour Occtax c'est dans `pr_occtax.defaults_nomenclatures_value`. Elles peuvent être définies pour chaque type de nomenclature ainsi que par organisme, règne et/ou `group2inpn`
- Si organisme = 0 alors la valeur par défaut s'applique à tous les organismes. Idem pour les règnes et `group2inpn`
- La fonction `pr_occtax.get_default_nomenclature_value` permet de renvoyer l'id de la nomenclature par défaut

- Ces valeurs par défaut sont aussi utilisées pour certains champs qui sont cachés (`statut_observation`, `floutage`, `statut_validation`...) mais ne sont donc pas modifiables par l'utilisateur
- Il existe aussi une table pour définir des valeurs par défaut générales de nomenclature (`ref_nomenclatures.defaults_nomenclatures_value`)
- Elles peuvent être administrées dans le module Admin de GeoNature

### 3.2.5 Métadonnées

- Elles sont gérées dans le schéma `gn_meta` basé sur le standard Métadonnées du SINP (<http://standards-sinp.mnhn.fr/category/standards/metadonnees/>)
- Elles permettent de gérer des jeux de données, des cadres d'acquisition, des acteurs (propriétaire, financeur, producteur...) et des protocoles
- Elles peuvent être administrées dans le module Métadonnées de GeoNature

### 3.2.6 Données SIG

- Le schéma `ref_geo` permet de gérer les données SIG (zonages, communes, MNT...) de manière centralisée, potentiellement partagé avec d'autres BDD
- Il contient une table des zonages, des types de zonages, des communes, des grilles (mailles) et un MNT raster ou vectorisé (<https://github.com/PnX-SI/GeoNature/issues/235>)
- La fonction `ref_geo.fct_get_area_intersection` permet de renvoyer les zonages intersectés par une observation en fournissant sa géométrie
- La fonction `ref_geo.fct_get_altitude_intersection` permet de renvoyer l'altitude min et max d'une observation en fournissant sa géométrie
- Les intersections d'une observation avec les zonages sont stockées au niveau de la synthèse (`gn_synthese.cor_area_synthese`) et non au niveau de la donnée source pour alléger et simplifier leur gestion

### 3.2.7 Profils de taxons

#### Introduction

GeoNature dispose d'un mécanisme permettant de calculer des profils pour chaque taxon en se basant sur les données validées présentes dans la Synthèse de l'instance.

Ces profils sont stockés dans un schéma dédié `gn_profiles`, et plus précisément dans les deux vues matérialisées suivantes :

1. La vue matérialisée `gn_profiles.vm_valid_profiles` comporte des informations générales sur chaque taxon :
  - L'aire d'occurrences
  - Les altitudes extrêmes d'observation du taxon
  - Les dates de première et de dernière observation
  - Le nombre de données valides pour le taxon considéré
2. La vue matérialisée `gn_profiles.vm_cor_taxon_phenology` comporte les "combinaisons" d'informations relatives à la phénologie des taxons (voir détail des calculs ci-dessous) :

- La période d’observation
- Le stade de vie (activable ou non)
- Les altitudes min et max
- Les altitudes “fiabiles” en écartant les valeurs extrêmes
- Le nombre de données correspondant à cette “combinaison phénologique”

La fonction `gn_profiles.refresh_profiles()` permet de rafraichir ces vues matérialisées.

Pour lancer manuellement cette fonction, ouvrez une console SQL et exécutez la requête `SELECT gn_profiles.refresh_profiles();`

Cette fonction est aussi disponible en tant que fonction GeoNature qu’il est préférable d’utiliser : `geonature_profiles update`

Pour automatiser l’exécution de cette fonction (tous les jours à minuit dans cet exemple), *créer une tâche planifiée*.

## Usage

Pour chaque taxon (`cd_ref`) disposant de données dans la vue `gn_profiles.v_synthese_for_profiles` (vue filtrée basée sur la synthèse de l’instance), un profil est généré. Il comporte l’aire d’occurrence, les limites altitudinales et les combinaisons phénologiques jugées cohérentes sur la base des données disponibles.

Ces profils sont déclinés sur :

- Le module de validation permet d’attirer l’attention des validateurs sur les données qui sortent du “cadre” déjà connu pour le taxon considéré, et d’apporter des éléments de contexte en complément de la donnée en cours de validation
- Le module Synthèse (fiche d’information, onglet validation) permet d’apporter des éléments de contexte en complément des données brutes consultées
- Le module Occtax permet d’alerter les utilisateurs lors de la saisie de données qui sortent du “cadre” déjà connu pour un taxon considéré

The screenshot displays the GeoNature2 validation interface. The main table lists validation records with the following columns: Details, Taxon, Date obs., Jeu de données, Observateur, and Cohérence. The records include various species like Apollon (L.), Parnassien apollon (L.), and others, with their respective observation dates and observers. A map on the right shows the geographical distribution of the records, with blue dots indicating locations in France and surrounding regions. The left sidebar contains filters for validation status, date, and location.

Détails	Taxon	Date obs.	Jeu de données	Observateur	Cohérence
<input type="checkbox"/>	Apollon (L.), Parnassien apollon (L.)	11-11-2020	Données opportunistes ;	Maillard Donovan	0/3
<input type="checkbox"/>	Apollon (L.), Parnassien apollon (L.)	09-11-2020	Données opportunistes ;	Maillard Donovan	2/3
<input type="checkbox"/>	Doublure jaune (L.)	17-06-2020	Données opportunistes ;	Guichard Grégory	3/3
<input type="checkbox"/>	Myrtil (L.), Myrtille (L.), Jurtine (L.), Janire (L.)	17-06-2020	Données opportunistes ;	Guichard Grégory	3/3
<input type="checkbox"/>	Demi-Deuil (L.), Échiquier (L.), Échiquier com	17-06-2020	Données opportunistes ;	Guichard Grégory	3/3
<input type="checkbox"/>	Mélie de Fruhstorfer (L.)	17-06-2020	Données opportunistes ;	Guichard Grégory	3/3
<input type="checkbox"/>	Fadet commun (L.), Procris (L.), Petit Papili	17-06-2020	Données opportunistes ;	Guichard Grégory	3/3
<input type="checkbox"/>	Celypha lacunata	16-06-2020	Données opportunistes ;	Francoz Philippe	2/3
<input type="checkbox"/>	Oncocera semirubella	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Acidalia sinuata (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Soyeuse (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Crambus uliginosellus	16-06-2020	Données opportunistes ;	Francoz Philippe	2/3
<input type="checkbox"/>	Brocatelle d'or (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Grand Nacré (L.), Aglaé (L.), Moyen-Nacré (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	2/3
<input type="checkbox"/>	Demi-Deuil (L.), Échiquier (L.), Échiquier com	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Chrysoteuchia culmella	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Myrtil (L.), Myrtille (L.), Jurtine (L.), Janire (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Azuré des Nerpuns (L.), Argus à bande noire	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Fadet commun (L.), Procris (L.), Petit Papili	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Piède du Lotier (L.), Piède de la Moutarde	16-06-2020	Données opportunistes ;	Francoz Philippe	2/3
<input type="checkbox"/>	Piède du Navet (L.), Papillon blanc veiné de	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Tristan (L.)	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Gazé (L.), Piède de l'Aubépine (L.), Piède	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3
<input type="checkbox"/>	Pterophorus pentadactylus	16-06-2020	Données opportunistes ;	Francoz Philippe	3/3



The screenshot shows the GeoNature2 web application interface. On the left is a sidebar with various filters and search options. The main content area is titled 'Validation' and contains several tabs. The 'Profil du taxon' tab is selected, showing a table of taxon information and a map of the distribution area. The table includes fields like 'Altitude minimale valide', 'Altitude maximale valide', 'Première observation valide', and 'Dernière observation valide'. The map shows the distribution area with a red dot indicating the current observation location.

Plusieurs fonctions permettent de vérifier si une donnée de la synthèse est cohérente au regard du profil du taxon en question :

- `gn_profiles.check_profile_distribution` : permet de vérifier si la donnée testée est totalement incluse dans l'aire d'occurrences déjà connue pour son taxon.
- `gn_profiles.check_profile_phenology` : permet de vérifier si la phénologie d'une donnée (période, stade de vie, altitudes) est une combinaison déjà connue dans le profil du taxon
- `gn_profiles.check_profile_altitudes` : permet de vérifier si une donnée est bien située dans la fourchette d'altitudes connue pour le taxon en question

## Configuration et paramétrage

### Paramètres de calcul des profils :

Le calcul des profils de taxons repose sur plusieurs variables, paramétrables soit pour tout le mécanisme, soit pour des taxons donnés.

Les paramètres généraux dans la table `gn_profiles.t_parameters` :

- Le paramètre `id_valid_status_for_profiles` : permet de lister les `id_nomenclatures` des statuts de validation à prendre en compte pour les calculs des profils. Par exemple, en ne listant que les identifiants des nomenclatures "Certain -très probable" et "Probable", seules ces données valides seront prises en compte lors du calcul des profils (comportement par défaut). En listant tous les identifiants des nomenclatures des statuts de validation, l'ensemble des données alimenteront les profils de taxons.
- Le paramètre `id_rang_for_profiles` : permet de lister les `id_rang` de Taxref à prendre en compte pour les calculs des profils. Par défaut, les profils ne sont calculés que pour les `cd_ref` correspondant à des Genres, Espèces et Sous-espèces.
- Le paramètre `proportion_kept_data` définit le pourcentage de données à conserver lors du calcul des altitudes valides (`gn_profiles.vm_cor_taxon_phenology`), en retirant ainsi les extrêmes. Ce paramètre, défini à 95% par défaut, doit être compris entre 51 et 100% (voir détails ci-après).

Les deux premiers paramètres permettent de filtrer les données dans la vue `gn_profiles.v_synthese_for_profiles`. Cette vue comporte les données de la synthèse qui répondent aux paramètres et

qui alimenteront les profils de taxons. Les clauses WHERE de cette vue peuvent être adaptées pour filtrer les données sur davantage de critères et répondre aux besoins plus spécifiques, mais sa structure doit rester inchangée.

Les paramètres définis par taxon le sont dans la table `gn_profiles.cor_taxons_profiles_parameters` :

Les profils peuvent être calculés avec des règles différentes en fonction des taxons. Ceux-ci sont définis au niveau du `cd_nom`, à n'importe quel rang (espèce, famille, règne etc). Ils seront appliqués de manière récursive à tous les taxons situés "sous" le `cd_ref` paramétré.

Dans le cas où un taxon hérite de plusieurs règles (une définie pour son ordre et une autre définie pour sa famille par exemple), les paramètres définis au plus proche du taxon considéré seront pris en compte.

Par exemple, s'il existe des paramètres pour le phylum "Animalia" (`cd_nom` 183716) et d'autres pour le renard (`cd_nom` 60585), les paramètres du renard seront appliqués en priorité pour cette espèce, mais les paramètres Animalia s'appliqueront à tous les autres animaux.

Les règles applicables à chaque taxon sont récupérées par la fonction `gn_profiles.get_profiles_parameters(cdnom)`.

Pour chaque `cd_nom`, il est ainsi possible de définir les paramètres suivants :

- `spatial_precision` : La précision spatiale utilisée pour calculer les profils. Elle est exprimée selon l'unité de mesure de la projection locale de l'instance GeoNature : mètres pour le Lambert93, degré pour le WGS84 etc. Elle définit à la fois la taille de la zone tampon appliquée autour de chaque observation pour définir l'aire d'occurrences du taxon, ainsi que la distance maximale admise entre le centroïde et les limites d'une observation pour qu'elle soit prise en compte lors du calcul des profils (évite qu'une donnée imprécise valide à elle seule une grande zone).
- `temporal_precision_days` : La précision temporelle en jours utilisée pour calculer les profils. Elle définit à la fois le pas de temps avec lequel la phénologie est calculée, ainsi que la précision temporelle minimale requise (différence entre date début et date fin de l'observation) pour qu'une donnée soit prise en compte dans le calcul des profils. Une précision de 365 jours ou plus permettra de ne pas tenir compte de la période (toutes les données seront dans une unique période de l'année).
- `active_life_stage` : Définit si le stade de vie doit être pris en compte ou non lors du calcul des profils.

Par défaut, une précision spatiale de 2000m et une précision spatiale de 10j (décade) sont paramétrés pour tous les phylums, sans tenir compte des stades de vie.

A terme, d'autres variables pourront compléter ces profils : habitats (`habref`) ou comportement (nidification, reproduction, migration...) notamment.

*Configuration - Activer/désactiver les profils :*

Il est possible de désactiver l'ensemble des fonctionnalités liées aux profils dans l'interface, en utilisant le paramètre suivant dans le fichier `geonature/config/geonature_config.toml`

```
[FRONTEND]
ENABLE_PROFILES = true/false
```

## Calcul des phénologies

Pour chaque taxon, la phénologie est calculée en croisant dans un premier temps les périodes d’observations et, selon les paramètres, les stades de vie.

Pour chacune des combinaisons obtenues (période x stade de vie), sont alors calculées :

- L’altitude minimale (toutes données comprises)
- L’altitude maximale (toutes données comprises)
- L’altitude minimale fiable (en retirant x% de données extrêmes selon le paramètre `proportion_kept_data`)
- L’altitude maximale fiable (en retirant x% de données extrêmes selon le paramètre `proportion_kept_data`)
- Le nombre de données valides correspondantes

### *Exclusion des données extrêmes*

Afin que des données exceptionnelles, bien que valides, ne soient pas considérées comme une “norme”, les profils permettent d’exclure un certain pourcentage de données extrêmes. Pour ce faire :

- Le nombre de données exclues est systématiquement arrondi à l’entier supérieur, pour les extrêmes “bas” et les extrêmes “hauts”
- Aucune altitude fiable n’est calculée s’il y a davantage de données exclues que de données conservées
- Le paramètre `proportion_kept_data` doit donc être compris entre 51 et 100% : en dessous de 50%, le nombre de données supprimées est supérieur au nombre de données conservées, aucune altitude fiable ne sera calculée. Si le paramètre est à 100%, les altitudes fiables seront identiques aux altitudes extrêmes observées pour la période (et le stade) donnés

Il faut donc  $(1/[1 - \text{proportion\_kept\_data}/100]) + 1$  données pour que des altitudes fiables soient calculées, soit :

- 101 données minimum par période/stade si `proportion_kept_data = 99`
- 51 données minimum par période/stade si `proportion_kept_data = 98`
- 21 données minimum par période/stade si `proportion_kept_data = 95`
- 11 données minimum par période/stade si `proportion_kept_data = 90`
- 3 données minimum par période/stade si `proportion_kept_data = 51`

## 3.2.8 Gestion de la sensibilité

### Introduction

Les règles de sensibilité définies par défaut sont issues des règles du SINP. Elles dépendent de l’espèce et de l’observation. C’est-à-dire que pour une espèce donnée, plusieurs niveaux de sensibilité sont possibles selon l’observation.

## Niveaux de sensibilité

Voici les 5 niveaux de sensibilité définis par le SINP :

- Sensible - Aucune diffusion
- Sensible - Diffusion au département
- Sensible - Diffusion à la maille 10km
- Sensible - Diffusion à la Commune ou Znieff
- Non sensible - Diffusion précise

Dans certains cas, des demandes consistent à rendre l'entiereté des observations d'une espèce (et donc une espèce) sensible. Cette documentation propose une méthode pour y arriver dans l'outil GeoNature.

## Pour plus d'informations

Vous pouvez consulter :

- La page du site du [MNHN traitant de la sensibilité](#).
- Le rapport de 2020 sur [La sensibilité des données du système d'information de l'inventaire du patrimoine naturel : méthodes, pratiques et usages](#) (J. Ichter et S. Robert).

## Attention

L'objectif de ce document n'est pas de modifier les règles établies par le SINP. Il est donc conseillé de respecter ces règles définies au niveau régional et national et donc de ne pas ajouter de règles locales.

## Intégration dans GeoNature

Le référentiel de sensibilité fournie par l'INPN est normalement intégré à GeoNature lors de son installation. Sinon, il peut être manuellement intégré avec la commande :

```
(venv)$ geonature db upgrade ref_sensitivity_inpn@head
```

## Schéma gn\_sensitivity

3 tables sont utilisées par ces fonctions :

- `t_sensitivity_rules` qui relie notamment une espèce à un niveau de sensibilité
- `cor_sensitivity_criteria` qui permet d'appliquer ce niveau de sensibilité en fonction d'un critère (par défaut, biologique)
- `cor_sensitivity_area` qui permet d'appliquer un niveau de sensibilité en fonction de la zone géographique (pas encore abordée ici)

S'il n'y a aucune entrée dans `cor_sensitivity_criteria`, le niveau de sensibilité défini dans `t_sensitivity_rules` est appliqué peu importe le statut biologique ou le comportement de l'occurrence. De même, s'il n'y a aucune entrée dans `cor_sensitivity_area`, le niveau de sensibilité est appliqué peu importe la localisation de l'observation.

## Schéma gn\_synthese

A chaque insertion d'une donnée dans la table `gn_synthese.synthese`, un trigger (`tri_insert_calculate_sensitivity`) fait appel à une fonction (`fct_tri_cal_sensitivity_on_each_statement`) qui appelle elle-même la fonction `gn_sensitivity.get_id_nomenclature_sensitivity` pour le calcul de la sensibilité.

La fonction `get_id_nomenclature_sensitivity` calcule le niveau de sensibilité en fonction de l'espèce, du type de sensibilité, de la durée de validité, de la période d'observation, du statut biologique et du comportement de l'occurrence.

## Personnalisation

Pour l'instant, seule la personnalisation de la sensibilité pour une espèce donnée (peu importe l'observation) est abordée ici.

## Sensibilité de l'espèce toute observation confondue

1. Dans `gn_sensitivity.t_sensitivity_rules` : Changez le niveau de sensibilité `id_nomenclature_sensitivity` par celui désiré. Pour la valeur à renseigner, voir dans `t_nomenclature` en filtrant avec `id_type=ref_nomenclatures.get_id_nomenclature_type('SENSIBILITE')`. En général l'identifiant varie entre 65 (non sensible) et 69 (aucune diffusion). Attention ces identifiants peuvent varier en fonction de votre installation.
2. Dans `cor_sensitivity_criteria` : s'il y a une correspondance d'`id_sensitivity` avec `t_sensitivity_rules`, modifiez ou supprimez cette ligne.
3. Lancez la commande SQL suivante :

```
REFRESH MATERIALIZED VIEW gn_sensitivity.t_sensitivity_rules_cd_ref;
```

Pour rafraîchir la vue matérialisée utilisée par les fonctions appelées par le trigger de `gn_synthese.synthese`.

4. Il est maintenant nécessaire de mettre à jour la sensibilité de vos observations présentes dans la synthèse. Pour cela, lancez la commande suivante :

```
(venv)$ geonature sensitivity update-synthese
```

Normalement, les valeurs dans la colonne `id_nomenclature_sensitivity` de la table `gn_synthese.synthese` ont changé. Vous pouvez le vérifier en navigant dans le module Synthèse puis dans les détails d'une observation de votre/vos espèce(s).

## 3.2.9 Fonctions

La base de données contient de nombreuses fonctions.

### gn\_synthese

Fonction	Paramètres	Ré-sultat	Description
get_default_nomenclature_value	idtype integer, id_nomenclature integer, id_organisme integer, regne character varying(20), group2_inpn character varying(255)	Entier	Function that return the default nomenclature id with a nomenclature type, organism id, regne, group2_inpn
fct_trig_insert_in_corona_synthese		Trigger	Trigger intersectant la géométrie d'une observation avec tous les zonages

**ref\_geo**

```
ref_geo.fct_get_altitude_intersection(IN mygeom geometry)
-- Fonction qui retourne l'altitude min et max de la géométrie passée en paramètre
```

```
ref_geo.fct_get_area_intersection(
  IN mygeom geometry,
  IN myidtype integer DEFAULT NULL::integer)
RETURNS TABLE(id_area integer, id_type integer, area_code character varying, area_
↳ name character varying)
-- Fonction qui retourne un tableau des zonages (id_area) intersectant la géométrie_
↳ passée en paramètre
```

```
ref_geo.get_id_area_type(mytype character varying) RETURNS integer
--Function which return the id_type_area from the type_code of an area type
```

**pr\_occtax**

```
pr_occtax.get_id_counting_from_id_releve(my_id_releve integer) RETURNS integer[]
-- Function which return the id_countings in an array (table pr_occtax.cor_counting_
↳ occtax) from the id_releve(integer)
```

```
get_default_nomenclature_value(mytype character varying, myidorganisme integer DEFAULT
↳ 0, myregne character varying(20) DEFAULT '0', mygroup2_inpn character varying(255)
↳ DEFAULT '0') RETURNS integer
--Function that return the default nomenclature id with wanteds nomenclature type,
↳ organisme id, regne, group2_inpn --Return -1 if nothing matche with given parameters
```

```
pr_occtax.insert_in_synthese(my_id_counting integer) RETURNS integer[]
```

**ref\_nomenclatures**

```
get_id_nomenclature_type(mytype character varying) RETURNS integer
--Function which return the id_type from the mnemonic of a nomenclature type
```

```
get_default_nomenclature_value(mytype character varying, myidorganisme integer DEFAULT
↳ 0) RETURNS integer
--Function that return the default nomenclature id with wanteds nomenclature type_
↳ (mnemonic), organisme id
--Return -1 if nothing matche with given parameters
```

```
check_nomenclature_type_by_mnemonic(id integer , mytype character varying) RETURNS
↳ boolean
--Function that checks if an id_nomenclature matches with wanted nomenclature type_
↳ (use mnemonic type)
```



```
check_nomenclature_type_by_cd_nomenclature(mycdnomenclature character varying ,
↳mytype character varying)
--Function that checks if an id_nomenclature matches with wanted nomenclature type
↳(use mnemonic type)
```

```
check_nomenclature_type_by_id(id integer, myidtype integer) RETURNS boolean
--Function that checks if an id_nomenclature matches with wanted nomenclature type
↳(use id_type)
```

```
get_id_nomenclature(
mytype character varying,
mycdnomenclature character varying)
RETURNS integer
--Function which return the id_nomenclature from an mnemonic_type and an cd_
↳nomenclature
```

```
get_nomenclature_label(
myidnomenclature integer,
mylanguage character varying
)
RETURNS character varying
--Function which return the label from the id_nomenclature and the language
```

```
get_cd_nomenclature(myidnomenclature integer) RETURNS character varying
--Function which return the cd_nomenclature from an id_nomenclature
```

```
get_filtered_nomenclature(mytype character varying, myregne character varying,
↳mygroup character varying)
RETURNS SETOF integer
--Function that returns a list of id_nomenclature depending on regne and/or group2_
↳inpn sent with parameters.
```

```
calculate_sensitivity(
mycdnom integer,
mynomenclatureid integer)
RETURNS integer
--Function to return id_nomenclature depending on observation sensitivity
--USAGE : SELECT ref_nomenclatures.calculate_sensitivity(240,21);
```

## gn\_profiles

```
gn_profiles.get_profiles_parameters(mycdnom integer)
RETURNS TABLE (cd_ref integer, spatial_precision integer, temporal_precision_days
↳integer, active_life_stage boolean, distance smallint)
-- fonction permettant de récupérer les paramètres les plus adaptés (définis au plus
↳proche du taxon) pour calculer le profil d'un taxon donné
-- par exemple, s'il existe des paramètres pour les "Animalia" des paramètres pour le
↳renard, les paramètres du renard surcoucheront les paramètres Animalia pour cette
↳espèce
```

```
gn_profiles.check_profile_distribution(
in_geom geometry,
profil_geom geometry
)
```

(continues on next page)

(continued from previous page)

**RETURNS** boolean

```
--fonction permettant de vérifier la cohérence d'une donnée d'occurrence en s
↳ 'assurant que sa localisation est totalement incluse dans l'aire d'occurrences
↳ valide définie par le profil du taxon en question
```

```
gn_profiles.check_profile_phenology(
    in_cd_ref integer,
    in_date_min date,
    in_date_max date,
    in_altitude_min integer,
    in_altitude_max integer,
    in_id_nomenclature_life_stage integer,
    check_life_stage boolean
)
```

**RETURNS** boolean

```
--fonction permettant de vérifier la cohérence d'une donnée d'occurrence en s
↳ 'assurant que sa phénologie (dates, altitude, stade de vie selon les paramètres)
↳ correspond bien à la phénologie valide définie par le profil du taxon en question
--La fonction renvoie 'false' pour les données trop imprécises (durée d'observation
↳ supérieure à la précision temporelle définie dans les paramètres des profils).
```

```
gn_profiles.check_profile_altitudes(
    in_alt_min integer,
    in_alt_max integer,
    profil_altitude_min integer,
    profil_altitude_max integer
)
```

**RETURNS** boolean

```
--fonction permettant de vérifier la cohérence d'une donnée d'occurrence en s
↳ 'assurant que son altitude se trouve entièrement comprise dans la fourchette
↳ altitudinale valide du taxon en question
```

### 3.2.10 Tables transversales

GeoNature contient aussi des tables de stockage transversales qui peuvent être utilisées par tous les modules. C'est le cas pour la validation, la sensibilité, l'historisation des modifications et les médias.

Cela permet de ne pas avoir à mettre en place des tables et mécanismes dans chaque module, mais de s'appuyer sur un stockage, des fonctions et développements factorisés, centralisés et partagés.

Ces tables utilisent notamment le mécanisme des UUID (identifiant unique) pour retrouver les enregistrements. Depuis une table source (Occtax ou un autre module) on peut retrouver les enregistrements stockés dans les tables transversales en utilisant un `WHERE <TABLE_TRANSVERSALE>.uuid_attached_row = <MON_UUID_SOURCE>` et ainsi retrouver l'historique de validation, les médias ou encore la sensibilité associés à une donnée.

Voir <https://github.com/PnX-SI/GeoNature/issues/339>

### 3.2.11 Triggers vers la synthèse

Voir ceux mis en place de Occtax vers Synthèse.

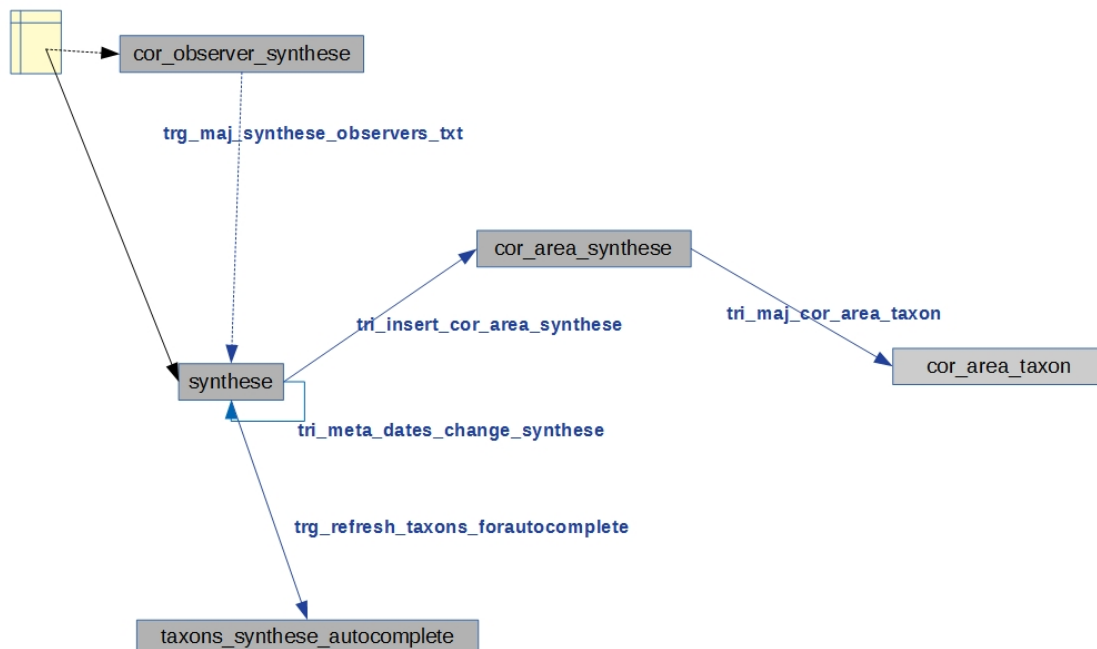
Cheminement d'une donnée Occtax :

1. Formulaire Occtax
2. Ecriture dans la table `cor_counting_occtax` et génération d'un nouvel UUID
3. Trigger d'écriture dans la table verticale `t_validations` à partir de la valeur par défaut de la nomenclature de validation (`gn_common.ref_nomenclatures.defaults_nomenclatures_value`)
4. Trigger d'écriture d'Occtax vers la synthèse (on ne maîtrise pas l'ordre de ces 2 triggers qui sont lancés en même temps)
5. Trigger de rapatriement du dernier statut de validation de la table verticale vers la synthèse.

### 3.2.12 Triggers dans la synthèse

Version 2.1.0 de GeoNature

INSERT data



#### Table : `synthese`

Table contenant l'ensemble des données. Respecte le standard Occurrence de taxon du SINP.

- `tri_meta_dates_change_synthese`
  - BEFORE INSERT OR UPDATE

- Mise à jour des champs `meta_create_date` et `meta_update_date`
- `tri_insert_cor_area_synthese`
  - AFTER INSERT OR UPDATE OF `the_geom_local`
  - Mise à jour de la table `cor_area_synthese`
  - Actions :
    1. Si update : suppression des enregistrements de la table `gn_synthese.cor_area_synthese` avec l'`id_synthese` concerné
    2. Insertion des `id_areas` intersectant la géométrie de la synthèse dans `gn_synthese.cor_area_synthese`. *Prise en compte de toutes les aires qu'elles soient ou non actives. Manque `enable = true`*
- `tri_del_area_synt_maj_corarea_tax`
  - BEFORE DELETE
  - Mise à jour des tables `cor_area_taxon` et `cor_area_synthese`
  - Actions :
    1. Récupération de l'ensemble des aires intersectant la donnée de synthèse
    2. Suppression des enregistrement de `cor_area_taxon` avec le `cd_nom` et les aires concernés
    3. Insertion dans `cor_area_taxon` recalculant les `max`, `nb_obs` et `couleur` pour chaque aire pour l'ensemble des données avec les aires concernées et le `cd_nom` concerné ne correspondant pas à la donnée supprimée
    4. Suppression des enregistrements de `gn_synthese.cor_area_synthese`
- `tri_update_cor_area_taxon_update_cd_nom`
  - AFTER UPDATE OF `cd_nom`
  - Mise à jour de la table `cor_area_taxon`
  - Actions :
    1. Récupération de l'ensemble des aires intersectant la donnée de synthèse
    2. Recalcul `cor_area_taxon` pour l'ancien `cd_nom` via fonction `gn_synthese.delete_and_insert_area_taxon`
    3. Recalcul `cor_area_taxon` pour le nouveau `cd_nom` via fonction `gn_synthese.delete_and_insert_area_taxon`

**Table : `cor_area_synthese`**

Table contenant l'ensemble des `id_areas` intersectant les enregistrements de la synthèse

- `tri_maj_cor_area_taxon`
  - AFTER INSERT OR UPDATE
  - Mise à jour des données de `cor_area_taxon`
  - Actions :
    1. Récupération du `cd_nom` en lien avec l'enregistrement `cor_area_synthese`
    2. Suppression des données de `cor_area_taxon` avec le `cd_nom` et `id_area` concernés
    3. Insertion des données dans `cor_area_taxon` en lien avec le `cd_nom` et `id_area`

**Table : cor\_observer\_synthese**

- `trg_maj_synthese_observers_txt`
  - AFTER INSERT OR UPDATE OR DELETE
  - Mise à jour du champ `observers` de la table `synthese`
  - Actions :
    1. Construction de la valeur textuelle des observateurs
    2. Mise à jour du champ `observer` de l'enregistrement de la table `synthese`

**FONCTIONS**

- `delete_and_insert_area_taxon`
  - Fonction qui met à jour la table `cor_area_taxon` en fonction d'un `cd_nom` et d'une liste d'`id_area`
  - Actions :
    1. Suppression des enregistrement de la table `cor_area_taxon` avec le `cd_nom` et les `id_area` concernés
    2. Insertion des données dans `cor_area_taxon`
- `color_taxon`
  - Fonction qui associe une couleur à une durée
  - *Passer les couleurs en paramètres : table `gn_commons.t_parameters` ?*
  - *Passer la fonction en immutable*

### 3.3 Modularité

Chaque module doit avoir son propre schéma dans la BDD, avec ses propres fichiers SQL de création comme le module OccTax : <https://github.com/PnX-SI/GeoNature/tree/develop/contrib/occtax/data>

Côté Backend, chaque module a aussi son modèle et ses routes : <https://github.com/PnX-SI/GeoNature/tree/develop/contrib/occtax/backend>

Idem côté Frontend, où chaque module a sa configuration et ses composants : <https://github.com/PnX-SI/GeoNature/tree/develop/contrib/occtax/frontend/app>

Mais en pouvant utiliser des composants du Cœur comme expliqué dans la documentation *Developpeur*.

Plus d'infos sur le développement d'un module : [https://github.com/PnX-SI/GeoNature/blob/master/docs/development.rst#d%C3%A9velopper-et-installer-un-gn\\_module](https://github.com/PnX-SI/GeoNature/blob/master/docs/development.rst#d%C3%A9velopper-et-installer-un-gn_module)

### 3.4 Configuration

Pour configurer GeoNature, actuellement il y a :

- Une configuration pour l'installation : `config/settings.ini`
- Une configuration globale de l'application : `<GEONATURE_DIRECTORY>/config/geonature_config.toml` (générée lors de l'installation de GeoNature)
- Une configuration par module : `<GEONATURE_DIRECTORY>/external_modules/<nom_module>/config/conf_gn_module.toml` (générée lors de l'installation d'un module)

- Une table `gn_commons.t_parameters` pour des paramètres gérés dans la BDD

## Administration GeoNature et son environnement

### Gestion applicative

#### Configuration de l'application



##### config

Configuration globale de l'application:

- Id connexion BDD,
- affichage/masquage champs formulaires
- paramétrage des exports

Nécessite exécution commande admin (ssh):  
- **`geonature update_configuration`**

#### Customisation de l'application



##### fontend/custom

Logo, CSS, composants

Pas de commande spécifique : FTP suffisant

#### Installation et configuration des modules



##### external\_modules

Nécessite exécution commande admin (ssh):  
- **`geonature install_gn_module`**  
- **`geonature update_configuration`**

### Gestion des référentiels (BDD)

#### Gestion des utilisateurs et droits



##### Application UsersHub

- Gestion centralisée des utilisateurs
- Gestion du CRUVED module par module

#### Gestion du référentiel Taxonomique



##### Application TaxHub

Gestion des taxons, listes, attributs et médias

#### Gestion des métadonnées



##### Module GeoNature-méta

Backoffice de gestion des métadonnées (cadres d'acquisition et jeux de données)  
En cours de dev

#### Gestion des nomenclatures



##### Sous-module nomenclature

Backoffice de gestion centralisée du vocabulaire applicatif  
Gestion des listes déroulantes

### 3.4.1 Configuration générale de l'application

L'installation de GeoNature génère le fichier de configuration globale `<GEONATURE_DIRECTORY>/config/geonature_config.toml`. Ce fichier est aussi copié dans le frontend (`frontend/conf/app.config.ts`), à ne pas modifier.

Par défaut, le fichier `<GEONATURE_DIRECTORY>/config/geonature_config.toml` est minimaliste et généré à partir des infos présentes dans le fichier `config/settings.ini`.

Il est possible de le compléter en surcouchant les paramètres présents dans le fichier `config/default_config.toml.example`.

A chaque modification du fichier global de configuration (`<GEONATURE_DIRECTORY>/config/geonature_config.toml`), il faut régénérer le fichier de configuration du frontend.

Ainsi après chaque modification des fichiers de configuration globale, placez-vous dans le backend de GeoNature (`/home/monuser/GeoNature/backend`) et lancez les commandes :

```
source venv/bin/activate
geonature update_configuration
sudo systemctl restart geonature
deactivate
```



### 3.4.2 Configuration d'un gn\_module

Lors de l'installation d'un module, un fichier de configuration est créé : `<MODULE_DIRECTORY>/config/conf_gn_module.toml`.

Comme pour la configuration globale, ce fichier est minimaliste et peut être surcouché. Le fichier `conf_gn_module.toml.example`, situé dans le répertoire `config` du module, décrit l'ensemble des variables de configuration disponibles ainsi que leurs valeurs par défaut.

A chaque modification de ce fichier, lancer les commandes suivantes depuis le backend de GeoNature (`/home/monuser/GeoNature/backend`). Le fichier est copié à destination du frontend `<nom_module>/frontend/app/module.config.ts`, qui est alors recompilé automatiquement.

```
source venv/bin/activate
geonature update_module_configuration <NOM_DE_MODULE>
deactivate
```

## 3.5 Exploitation

### 3.5.1 Logs

- Logs d'installation de GeoNature : `geonature/install/install.log`
- Logs de GeoNature : `/var/log/geonature.log`
- Logs de TaxHub : `/var/log/taxhub.log`
- Logs de UsersHub : `/var/log/usershub.log`

### 3.5.2 Commandes GeoNature

GeoNature est fourni avec une série de commandes pour administrer l'application. Pour les exécuter, il est nécessaire d'être dans le virtualenv python de GeoNature

```
cd <GEONATURE_DIRECTORY>/backend
source venv/bin/activate
```

Le préfixe (venv) se met alors au début de votre invite de commande.

Voici la liste des commandes disponibles (aussi disponibles en tapant la commande `geonature --help`):

- `activate_gn_module` : Active un `gn_module` installé (Possibilité d'activer seulement le backend ou le frontend)
- `deactivate_gn_module` : Désactive `gn_un module activé` (Possibilité de désactiver seulement le backend ou le frontend)
- `dev_back` : Lance le backend en mode développement
- `dev_front` : Lance le frontend en mode développement
- `generate_frontend_module_route` : Génère ou régénère le fichier de routing du frontend en incluant les `gn_module` installés (Fait automatiquement lors de l'installation d'un module)
- `install_gn_module` : Installe un `gn_module`
- `update_configuration` : Met à jour la configuration du cœur de l'application. A exécuter suite à une modification du fichier `geonature_config.toml`

- `update_module_configuration` : Met à jour la configuration d'un module. A exécuter suite à une modification du fichier `conf_gn_module.toml`.

Effectuez `geonature <nom_commande> --help` pour accéder à la documentation et à des exemples d'utilisation de chaque commande.

### 3.5.3 Démarrer / arrêter les API

- Démarrer GeoNature : `systemctl start geonature`
- Arrêter GeoNature : `systemctl stop geonature`
- Redémarrer GeoNature : `systemctl restart geonature`
- Vérifier l'état de GeoNature : `systemctl status geonature`

Les mêmes commandes sont disponibles pour TaxHub en remplaçant `geonature` par `taxhub`.

### 3.5.4 Supervision des services

- Vérifier que les applications GeoNature et TaxHub sont accessibles en http
- Vérifier que leurs services (API) sont lancés et fonctionnent correctement (tester les deux routes ci-dessous).
  - Exemple de route locale pour tester l'API GeoNature : `http://127.0.0.1:8000/occtax/defaultNomenclatures` qui ne doit pas renvoyer de 404. URL absolue : `https://urlgeonature/api/occtax/defaultNomenclatures`
  - Exemple de route locale pour tester l'API TaxHub : `http://127.0.0.1:5000/api/taxref/regnewithgroupe2` qui ne doit pas renvoyer de 404. URL absolue : `https://urltaxhub/api/taxref/regnewithgroupe2`
- Vérifier que les fichiers de logs de TaxHub et GeoNature ne sont pas trop volumineux pour la capacité du serveur
- Vérifier que les services nécessaires au fonctionnement de l'application tournent bien (Apache, PostgreSQL)

### 3.5.5 Maintenance

Lors d'une opération de maintenance (montée en version, modification de la base de données...), vous pouvez rendre l'application momentanément indisponible.

Pour cela, désactivez la configuration Apache de GeoNature, puis activez la configuration du mode de maintenance :

```
sudo a2dissite geonature
sudo a2ensite geonature_maintenance
sudo apachectl restart
```

A la fin de l'opération de maintenance, effectuer la manipulation inverse :

```
sudo a2dissite geonature_maintenance
sudo a2ensite geonature
sudo apachectl restart
```

Attention : ne pas stopper le backend (des opérations en BDD en cours pourraient être corrompues)

- Redémarrage de PostgreSQL

Si vous effectuez des manipulations de PostgreSQL qui nécessitent un redémarrage du SGBD (`sudo service postgresql restart`), il faut impérativement lancer un redémarrage des API GeoNature et TaxHub pour que celles-ci continuent de fonctionner. Pour cela, lancez les commandes `sudo systemctl restart geonature` et `sudo systemctl restart taxhub` (GeoNature 2.8+).

**NB:** Ne pas faire ces manipulations sans avertir les utilisateurs d'une perturbation temporaire des applications.

### 3.5.6 Paramètres Gunicorn

Voici les paramètres de Gunicorn par défaut :

- GUNICORN\_PROC\_NAME=geonature
- GUNICORN\_NUM\_WORKERS=4
- GUNICORN\_HOST=127.0.0.1
- GUNICORN\_PORT=8000
- GUNICORN\_TIMEOUT=30

Pour modifier une de ces variables, créer un fichier environ à la racine de votre dossier GeoNature, et indiquer la variable d'environnement avec sa valeur modifiée.

Si vous souhaitez modifier de manière plus avancé la ligne de commande gunicorn, surcouchez le service systemd :

- Lancez `sudo systemctl edit geonature` ce qui va créer le fichier `/etc/systemd/system/geonature.service.d/override.conf` et ouvrir un éditeur pour vous permettre de le modifier
- Indiquez :

```
[Service]
ExecStart=
ExecStart=/path/to/venv/bin/unicorn geonature:create_app() ...
```

Note : le premier ExecStart permet de réinitialiser la commande de lancement de gunicorn.

## 3.6 Sauvegarde et restauration

### 3.6.1 Sauvegarde

- Sauvegarde de la base de données :

Les sauvegardes de la BDD sont à faire avec l'utilisateur postgres. Commencer par créer un répertoire et lui donner des droits sur le répertoire où seront faites les sauvegardes.

```
$ # Créer le répertoire pour stocker les sauvegardes
$ mkdir /home/~whoami~/backup
$ # Ajouter l'utilisateur postgres au groupe de l'utilisateur linux courant pour qu
↪ 'il ait les droits d'écrire dans les mêmes répertoires
$ sudo adduser postgres ~whoami~
$ # ajout de droit aux groupes de l'utilisateur courant sur le répertoire `backup`
$ chmod g+rxw /home/~whoami~/backup
```

Connectez-vous avec l'utilisateur linux postgres pour lancer une sauvegarde de la BDD :

```
sudo su postgres
pg_dump -Fc geonature2db > /home/~whoami~/backup/~date +%Y-%m-%d-%H:%M~-geonaturedb.
↪ backup
exit
```

Si la sauvegarde ne se fait pas, c'est qu'il faut revoir les droits du répertoire où sont faites les sauvegardes pour que l'utilisateur `postgres` puisse y écrire

Opération à faire régulièrement grâce à une tâche cron.

- Sauvegarde des fichiers de configuration :

```
cd /home/~whoami~/geonature/config
tar -zcvf /home/~whoami~/backup/~date +%Y%m%d%H%M~-geonature_config.tar.gz ./
```

Opération à faire à chaque modification d'un paramètre de configuration.

- Sauvegarde des fichiers de customisation :

```
cd /home/~whoami~/geonature/frontend/src/custom
tar -zcvf /home/~whoami~/~date +%Y%m%d%H%M~-geonature_custom.tar.gz ./
```

Opération à faire à chaque modification de la customisation de l'application.

- Sauvegarde des modules externes :

```
cd /home/~whoami~/geonature/external_modules
tar -zcvf /home/~whoami~/backup/~date +%Y%m%d%H%M~-external_modules.tar.gz ./
```

### 3.6.2 Restauration

- Restauration de la base de données :

- Créer une base de données vierge (on part du principe que la base de données `geonature2db` n'existe pas ou plus). Sinon adaptez le nom de la BDD et également la configuration de connexion de l'application à la BDD dans `<GEONATURE_DIRECTORY>/config/geonature_config.toml`

```
sudo -n -u postgres -s createdb -O <MON_USER> geonature2db
sudo -n -u postgres -s psql -d geonature2db -c "CREATE EXTENSION IF NOT_
↳ EXISTS postgis;"
sudo -n -u postgres -s psql -d geonature2db -c "CREATE EXTENSION IF NOT_
↳ EXISTS hstore;"
sudo -n -u postgres -s psql -d geonature2db -c 'CREATE EXTENSION IF NOT_
↳ EXISTS "uuid-oss";'
sudo -n -u postgres -s psql -d geonature2db -c "CREATE EXTENSION IF NOT_
↳ EXISTS pg_trgm;"
sudo -n -u postgres -s psql -d geonature2db -c "CREATE EXTENSION IF NOT_
↳ EXISTS postgis_raster;" # postgis>=3.0 (Debian 11)
```

- Restaurer la BDD à partir du backup

```
sudo su postgres
pg_restore -d geonature2db <MY_BACKUP_DIRECTORY_PATH>/201803150917-
↳ geonaturedb.backup
```

- Restauration de la configuration et de la customisation :

- Décompresser les fichiers précédemment sauvegardés pour les remettre au bon emplacement :

```
sudo rm <GEONATURE_DIRECTORY>/config/*
cd <GEONATURE_DIRECTORY>/config
sudo tar -zxvf <MY_BACKUP_DIRECTORY>/201803150953-geonature_config.tar.gz
```

(continues on next page)

(continued from previous page)

```
cd /home/<MY_USER>/geonature/frontend/src/custom
rm -r <MY_USER>/geonature/frontend/src/custom/*
tar -zxvf <MY_BACKUP_DIRECTORY>/201803150953-geonature_custom.tar.gz

rm /home/<MY_USER>/geonature/external_modules/*
cd <GEONATURE_DIRECTORY>/external_modules
tar -zxvf <MY_BACKUP_DIRECTORY>/201803151036-external_modules.tar.gz
```

- Relancer l'application GeoNature

## 3.7 Customisation

La customisation de l'application nécessite de relancer la compilation du frontend à chaque modification. Cette opération étant relativement longue, une solution alternative (mais avancée) consiste à passer le frontend de manière temporaire en mode 'développement'.

Pour cela exécuter la commande suivante depuis le répertoire `frontend`

```
npm run start -- --host=0.0.0.0 --disable-host-check
```

L'application est désormais disponible sur un serveur de développement à la même adresse que précédemment, mais sur le port 4200 : <http://test.geonature.fr:4200>

Ouvrez un nouveau terminal (pour laisser tourner le serveur de développement), puis modifier la variable `URL_APPLICATION` dans le fichier `geonature_config.toml` en mettant l'adresse ci-dessus et relancer l'application (`sudo supervisorctl restart geonature2` ou `sudo systemctl restart geonature`)

A chaque modification d'un fichier du frontend, une compilation rapide est relancée et votre navigateur se rafraichit automatiquement en intégrant les dernières modifications.

Une fois les modifications terminées, remodifier le fichier `geonature_config.toml` pour remettre l'URL initiale, relancez l'application (`sudo supervisorctl restart geonature2` ou `sudo systemctl restart geonature`), puis relancez la compilation du frontend (`npm run build`). Faites enfin un `ctrl+c` dans le terminal où le frontend a été lancé pour stopper le serveur de développement.

Si la manipulation vous paraît compliquée, vous pouvez suivre la documentation qui suit, qui fait relancer la compilation du frontend à chaque modification.

### 3.7.1 Intégrer son logo

Le logo affiché dans la barre de navigation de GeoNature peut être modifié dans le répertoire `geonature/frontend/src/custom/images`. Remplacez alors le fichier `logo_structure.png` par votre propre logo, en conservant ce nom pour le nouveau fichier. Le bandeau fait 50px de hauteur, vous pouvez donc mettre une image faisant cette hauteur. Il est également possible de modifier la taille de l'image en CSS dans le fichier `frontend/src/assets/custom.css` de la manière suivante:

```
/* la balise img affichant l'image a l'id 'logo-structure */
#logo-structure {
  height: 50px;
  width: 80px;
}
```

Relancez la construction de l'interface :

```
cd /home/~whoami~/geonature/frontend
npm run build
```

### 3.7.2 Customiser le contenu

- Customiser le contenu de la page d'introduction :

Le texte d'introduction et le titre de la page d'Accueil de GeoNature peuvent être modifiés à tout moment, sans réinstallation de l'application. Il en est de même pour le bouton d'accès à la synthèse.

Il suffit pour cela de mettre à jour le fichier `introduction.component.html`, situé dans le répertoire `geonature/frontend/src/custom/components/introduction`.

Afin que ces modifications soient prises en compte dans l'interface, il est nécessaire de relancer les commandes suivantes :

```
cd /home/~whoami~/geonature/frontend
npm run build
```

- Customiser le contenu du pied de page :

Le pied de page peut être customisé de la même manière, en renseignant le fichier `footer.component.html`, situé dans le répertoire `geonature/frontend/src/custom/components/footer`

De la même manière, il est nécessaire de relancer les commandes suivantes pour que les modifications soient prises en compte :

```
cd /home/~whoami~/geonature/frontend
npm run build
```

### 3.7.3 Customiser l'aspect esthétique

Les couleurs de textes, couleurs de fonds, forme des boutons etc peuvent être adaptées en renseignant le fichier `custom.css`, situé dans le répertoire `geonature/frontend/src/assets`.

Pour remplacer la couleur de fond du bandeau de navigation par une image, on peut par exemple apporter la modification suivante :

```
html body pnx-root pnx-nav-home mat-sidenav-container.sidenav-container.mat-drawer-
↪container.mat-sidenav-container mat-sidenav-content.mat-drawer-content.mat-sidenav-
↪content mat-toolbar#app-toolbar.row.mat-toolbar
{
  background :
    url(bandeau_test.jpg)
}
```

Dans ce cas, l'image `bandeau_test.jpg` doit se trouver dans le répertoire `geonature/frontend/src`.

Comme pour la modification des contenus, il est nécessaire de relancer la commande suivante pour que les modifications soient prises en compte :

```
cd /home/~whoami~/geonature/frontend
npm run build
```



### 3.7.4 Customiser les noms et pictos des modules

Vous pouvez modifier l'intitulé et le pictogramme des modules dans le menu des modules. Pour cela, adaptez le contenu des champs `module_label` et `module_picto` (avec des icônes de la librairie Font Awesome - <https://fontawesome.com>) dans la table `gn_commons.t_modules`.

Exemple :

```
-- Module Occtax
UPDATE gn_commons.t_modules SET module_label = 'Occtax' WHERE module_code = 'OCCTAX';
UPDATE gn_commons.t_modules SET module_picto = 'fa-map-marker' WHERE module_code =
↳ 'OCCTAX';
-- Module Occhab
UPDATE gn_commons.t_modules SET module_label = 'Occhab' WHERE module_code = 'OCCHAB';
UPDATE gn_commons.t_modules SET module_picto = 'fa-leaf' WHERE module_code = 'OCCHAB';
-- Module Import
UPDATE gn_commons.t_modules SET module_label = 'Import' WHERE module_code = 'IMPORT';
UPDATE gn_commons.t_modules SET module_picto = 'fa-upload' WHERE module_code = 'IMPORT
↳ ';
-- Module Export
UPDATE gn_commons.t_modules SET module_label = 'Export' WHERE module_code = 'EXPORTS';
UPDATE gn_commons.t_modules SET module_picto = 'fa-download' WHERE module_code =
↳ 'EXPORTS';
-- Module Dashboard
UPDATE gn_commons.t_modules SET module_label = 'Dashboard' WHERE module_code =
↳ 'DASHBOARD';
UPDATE gn_commons.t_modules SET module_picto = 'fa-bar-chart' WHERE module_code =
↳ 'DASHBOARD';
-- Module Validation
UPDATE gn_commons.t_modules SET module_label = 'Validation' WHERE module_code =
↳ 'VALIDATION';
UPDATE gn_commons.t_modules SET module_picto = 'fa-check' WHERE module_code =
↳ 'VALIDATION';
-- Module Monitorings (Suivis)
UPDATE gn_commons.t_modules SET module_label = 'Suivis' WHERE module_code =
↳ 'MONITORINGS';
UPDATE gn_commons.t_modules SET module_picto = 'fa-eye' WHERE module_code =
↳ 'MONITORINGS';
```

Depuis la version 2.5.0, il est aussi possible de customiser l'ordre des modules dans le menu, par ordre alphabétique par défaut, en renseignant le champs `gn_commons.t_modules.module_order`.

### 3.7.5 Customiser les exports PDF

Vous pouvez modifier le bandeau et le logo fournis par défaut dans les exports PDF en modifiant les images `Bandeau_pdf.png` et `Logo_pdf.png` dans `backend/static/images`. Le style des fichiers est également customisable grâce au fichier “`backend/geonature/static/css/custom.css`”. La classe `main-color` permet notamment de changer la couleur des séparateurs (orange par défaut).

## 3.8 Intégrer des données

### 3.8.1 Référentiel géographique

GeoNature est fourni avec des données géographiques de base sur la métropole (MNT national à 250m et communes de métropole).

1. Si vous souhaitez modifier le MNT pour mettre celui de votre territoire :

- Videz le contenu des tables `ref_geo.dem` et éventuellement `ref_geo.dem_vector`
- Uploadez le(s) fichier(s) du MNT sur le serveur
- Suivez la procédure de chargement du MNT en l’adaptant : \* [https://github.com/PnX-SI/GeoNature/blob/master/backend/geonature/migrations/versions/1715cf31a75d\\_insert\\_ign\\_250m\\_bd\\_alti\\_in\\_dem.py](https://github.com/PnX-SI/GeoNature/blob/master/backend/geonature/migrations/versions/1715cf31a75d_insert_ign_250m_bd_alti_in_dem.py) \* [https://github.com/PnX-SI/GeoNature/blob/master/backend/geonature/migrations/versions/87651375c2e8\\_vectorize\\_ign\\_bd\\_alti.py](https://github.com/PnX-SI/GeoNature/blob/master/backend/geonature/migrations/versions/87651375c2e8_vectorize_ign_bd_alti.py)

*TODO : Procédure à améliorer et simplifier : <https://github.com/PnX-SI/GeoNature/issues/235>*

Si vous n’avez pas choisi d’intégrer le raster MNT national à 250m fourni par défaut lors de l’installation ou que vous souhaitez le remplacer, voici les commandes qui vous permettront de le faire.

Suppression du MNT par défaut (adapter le nom de la base de données : `MYDBNAME`).

```
sudo -n -u postgres -s psql -d MYDBNAME -c "TRUNCATE TABLE ref_geo.dem;"
sudo -n -u postgres -s psql -d MYDBNAME -c "TRUNCATE TABLE ref_geo.dem_vector;"
```

Placer votre propre fichier MNT (ou vos différents fichiers “dalles”) dans le répertoire `/tmp/geonature` (adapter le nom du fichier et son chemin ainsi que les paramètres en majuscule).

Pour utiliser celui proposé par défaut :

```
wget --cache=off http://geonature.fr/data/ign/BDALTIV2_2-0_250M_ASC_LAMB93-IGN69_
↳FRANCE_2017-06-21.zip -P /tmp/geonature
unzip /tmp/geonature/BDALTIV2_2-0_250M_ASC_LAMB93-IGN69_FRANCE_2017-06-21.zip -d /tmp/
↳geonature
export PGPASSWORD=MYUSERPGPASS;raster2pgsql -s MYSRID -c -C -I -M -d -t 5x5 /tmp/
↳geonature/BDALTIV2_250M_FXX_0098_7150_MNT_LAMB93-IGN69.asc ref_geo.dem|psql -h
↳localhost -U MYPGUSER -d MYDBNAME
sudo -n -u postgres -s psql -d MYDBNAME -c "REINDEX INDEX ref_geo.dem_st_convexhull_
↳idx;"
```

Si votre MNT source est constitué de plusieurs fichiers (dalles), assurez vous que toutes vos dalles ont le même système de projection et le même format de fichier (tiff, asc, ou img par exemple). Après avoir chargé vos fichiers dans `tmp/geonature` (par exemple), vous pouvez lancer la commande `export` en remplaçant le nom des fichiers par `*.asc` :

```
export PGPASSWORD=MYUSERPGPASS;raster2pgsql -s MYSRID -c -C -I -M -d -t 5x5 /tmp/
↳geonature/*.asc ref_geo.dem|psql -h localhost -U MYPGUSER -d MYDBNAME
```

Si vous souhaitez vectoriser le raster MNT pour de meilleures performances lors des calculs en masse de l'altitude à partir de la localisation des observations, vous pouvez le faire en lançant les commandes ci-dessous. Sachez que cela prendra du temps et beaucoup d'espace disque (2.8Go supplémentaires environ pour le fichier DEM France à 250m).

```
sudo -n -u postgres -s psql -d MYDBNAME -c "INSERT INTO ref_geo.dem_vector (geom,
↪val) SELECT (ST_DumpAsPolygons(rast)).* FROM ref_geo.dem; "
sudo -n -u postgres -s psql -d MYDBNAME -c "REINDEX INDEX ref_geo.index_dem_vector_
↪geom; "
```

Si `ref_geo.dem_vector` est remplie, cette table est utilisée pour le calcul de l'altitude à la place de la table `ref_geo.dem`

2. Si vous souhaitez modifier ou ajouter des zonages administratifs, réglementaires ou naturels :

- Vérifiez que leur type existe dans la table `ref_geo.bib_areas_types`, sinon ajoutez-les
- Ajoutez vos zonages dans la table `ref_geo.l_areas` en faisant bien référence à un `id_type` de `ref_geo.bib_areas_types`. Vous pouvez faire cela en SQL ou en faisant des copier/coller de vos zonages directement dans QGIS
- Pour les grilles et les communes, vous pouvez ensuite compléter leurs tables d'extension `ref_geo.li_grids` et `ref_geo.li_municipalities`

### 3.8.2 Données externes

Il peut s'agir de données partenaires, de données historiques ou de données saisies dans d'autres outils.

2 possibilités s'offrent à vous :

- Créer un schéma dédié aux données pour les intégrer de manière complète et en extraire les DEE dans la Synthèse
- N'intégrer que les DEE dans la Synthèse

Nous présenterons ici la première solution qui est privilégiée pour disposer des données brutes mais aussi les avoir dans la Synthèse.

- Créer un JDD dédié (`gn_meta.t_datasets`) ou utilisez-en un existant. Eventuellement un CA si elles ne s'intègrent pas dans un CA déjà existant.
- Ajouter une Source de données dans `gn_synthese.t_sources` ou utilisez en une existante.
- Créer le schéma dédié à accueillir les données brutes.
- Créer les tables nécessaires à accueillir les données brutes.
- Intégrer les données dans ces tables (avec les fonctions de `gn_imports`, avec QGIS ou pgAdmin).
- Pour alimenter la Synthèse à partir des tables sources, vous pouvez mettre en place des triggers (en s'inspirant de ceux de OccTax) ou bien faire une requête spécifique si les données sources ne sont plus amenées à évoluer.

Pour des exemples plus précis, illustrées et commentées, vous pouvez consulter les 2 exemples d'import dans cette documentation (Import niveau et Import niveau 2).

Vous pouvez aussi vous inspirer des exemples avancés de migration des données de GeoNature V1 vers GeoNature V2 : <https://github.com/PnX-SI/GeoNature/tree/master/data/migrations/v1tov2>

- Import depuis SICEN (ObsOcc) : <https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/sicen>
- Import depuis SERENA : <https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/serena>

- Import continu : <https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/generic>
- Import d'un CSV historique (Flavia) : <https://github.com/PnX-SI/Ressources-techniques/blob/master/GeoNature/V2/2018-12-csv-vers-synthese-FLAVIA.sql>

## 3.9 Création de compte

### 3.9.1 Configuration de la création de compte

Depuis la version 2.1.0, UsersHub propose une API de création de compte utilisateur. Une interface a été ajoutée à GeoNature pour permettre aux futurs utilisateurs de faire des demandes de création de compte depuis la page d'authentification de GeoNature. Ce mode est activable/désactivable depuis la configuration globale de GeoNature.

Pour des raisons de sécurité, l'API de création de compte est réservée aux utilisateurs "admin" grâce à un token secret. GeoNature a donc besoin de se connecter en tant qu'administrateur à UsersHub pour exécuter les requêtes d'administration de compte. Renseigner les paramètres suivants dans le fichier de configuration (`geonature_config.toml`). L'utilisateur doit avoir des droits 6 dans UsersHub

```
[USERSHUB]
URL_USERSHUB = 'http://mon_adresse_usershub.fr' # sans slash final
# Administrateur de mon application
ADMIN_APPLICATION_LOGIN = "login_admin_usershub"
ADMIN_APPLICATION_PASSWORD = "password_admin_usershub"
```

Les fonctionnalités de création de compte nécessitent l'envoi d'emails pour vérifier l'identité des demandeurs de compte. Il est donc nécessaire d'avoir un serveur SMTP capable d'envoyer des emails. Renseigner la rubrique `MAIL_CONFIG` de la configuration. La description détaillée des paramètres de configuration d'envoi des emails est disponible dans la [documentation de Flask-Mail](#). Exemple :

```
[MAIL_CONFIG]
MAIL_SERVER = 'mail.espaces-naturels.fr'
MAIL_PORT = 465
MAIL_USE_TLS = false
MAIL_USE_SSL = true
MAIL_USERNAME = 'mon_email@email.io'
MAIL_PASSWORD = 'monpassword'
MAIL_DEFAULT_SENDER = 'mon_email@email.io'
MAIL_ASCII_ATTACHMENTS = false
```

Pour activer cette fonctionnalité (qui est par défaut désactivée), modifier le fichier de configuration de la manière suivante :

NB : tous les paramètres décrits ci-dessous doivent être dans la rubrique `[ACCOUNT_MANAGEMENT]`

```
[ACCOUNT_MANAGEMENT]
ENABLE_SIGN_UP = true
```

Deux modes sont alors disponibles. Soit l'utilisateur est automatiquement accepté et un compte lui est créé après une confirmation de son email, soit un mail est envoyé à un administrateur pour confirmer la demande. Le compte ne sera créé qu'après validation par l'administrateur. Le paramètre `AUTO_ACCOUNT_CREATION` contrôle ce comportement (par défaut le compte créé sans validation par un administrateur: `true`). Dans le mode "création de compte validé par administrateur", il est indispensable de renseigner un email où seront envoyés les emails de validation (paramètre `VALIDATOR_EMAIL`)

```
# automatique
[ACCOUNT_MANAGEMENT]
    ENABLE_SIGN_UP = true
    AUTO_ACCOUNT_CREATION = true

# validé par admin
[ACCOUNT_MANAGEMENT]
    ENABLE_SIGN_UP = true
    AUTO_ACCOUNT_CREATION = false
    VALIDATOR_EMAIL = 'email@validateur.io'
```

L'utilisateur qui demande la création de compte est automatiquement mis dans un "groupe" UsersHub (par défaut, il s'agit du groupe "En poste"). Ce groupe est paramétrable depuis la table utilisateurs.cor\_role\_app\_profil. (La ligne où is\_default\_group\_for\_app = true sera utilisée comme groupe par défaut pour GeoNature). Il n'est pas en paramètre de GeoNature puisqu'il serait falsifiable via l'API. **Attention**, si vous effectuez une migration depuis une version de GeoNature < 2.2.0, aucun groupe par défaut n'est défini, vous devez définir à la main le groupe par défaut pour l'application GeoNature dans la table utilisateurs.cor\_role\_app\_profil.

Dans le mode "création de compte validé par administrateur", lorsque l'inscription est validée par un administrateur, un email est envoyé à l'utilisateur pour lui indiquer la confirmation de son inscription. Il est possible de personnaliser le texte de la partie finale de cet email située juste avant la signature à l'aide du paramètre ADDON\_USER\_EMAIL (toujours à ajouter à la rubrique [ACCOUNT\_MANAGEMENT]). Vous pouvez utiliser des balises HTML compatibles avec les emails pour ce texte.

```
[ACCOUNT_MANAGEMENT]
    ADDON_USER_EMAIL = """<p>
        Toute l'équipe de GeoNature vous remercie pour votre inscription.
    </p>"""
```

Il est également possible de créer automatiquement un jeu de données et un cadre d'acquisition "personnel" à l'utilisateur afin qu'il puisse saisir des données dès sa création de compte via le paramètre AUTO\_DATASET\_CREATION. Par la suite l'administrateur pourra rattacher l'utilisateur à des JDD et CA via son organisme.

```
[ACCOUNT_MANAGEMENT]
    AUTO_ACCOUNT_CREATION = true
    ENABLE_SIGN_UP = true
    AUTO_DATASET_CREATION = true
```

### 3.9.2 Customisation du formulaire

Le formulaire de création de compte est par défaut assez minimaliste (nom, prénom, email, mot de passe, organisme, remarque).

*NB* l'organisme est demandé à l'utilisateur à titre "informatif", c'est à l'administrateur de rattacher individuellement l'utilisateur à son organisme, et éventuellement de le créer, s'il n'existe pas.

Il est possible d'ajouter des champs au formulaire grâce à un générateur contrôlé par la configuration. Plusieurs type de champs peuvent être ajoutés (text, textarea, number, select, checkbox mais aussi taxonomy, nomenclature etc...).

L'exemple ci-dessous permet de créer un champs de type "checkbox" obligatoire, avec un lien vers un document (une charte par exemple) et un champ de type "select", non obligatoire. (voir le fichier config/geonature\_config.toml.example pour un exemple plus exhaustif).

```
[ACCOUNT_MANAGEMENT]
[[ACCOUNT_MANAGEMENT.ACCOUNT_FORM]]
    type_widget = "checkbox"
    attribut_label = ""
    <a target="_blank" href="http://docs.geonature.fr">
        J'ai lu et j'accepte la charte
    </a>""
    attribut_name = "validate_charte"
    values = [true]
    required = true

[[ACCOUNT_MANAGEMENT.ACCOUNT_FORM]]
    type_widget = "select"
    attribut_label = "Exemple select"
    attribut_name = "select_test"
    values = ["value1", "value2"]
    required = false
```

### 3.9.3 Espace utilisateur

Enfin, un espace “utilisateur” est accessible lorsque l’on est connecté, permettant de modifier ses informations personnelles, y compris son mot de passe.

Cet espace est activable grâce au paramètre `ENABLE_USER_MANAGEMENT`. Par défaut, il est désactivé.

```
[ACCOUNT_MANAGEMENT]
AUTO_ACCOUNT_CREATION = true
ENABLE_SIGN_UP = true
ENABLE_USER_MANAGEMENT = true
```

## 3.10 Accès public

Cette section de la documentation concerne l’implémentation d’un utilisateur-lecteur pour votre instance GeoNature.

Étapes :

#### 1/ UsersHub :

- Aller dans la section *Utilisateurs*
- Créer un utilisateur
- Définir un identifiant et un mot de passe (par défaut utilisateur ‘public’ et mot de passe ‘public’)
- Aller ensuite dans la section *Applications*
- Pour GeoNature, cliquer sur le premier icône ‘Voir les membres’
- Cliquer sur ajouter un rôle
- Choisir l’utilisateur juste créé
- Attribuer le rôle 1, ‘lecteur’

#### 2/ Configuration GeoNature :

- Reporter identifiant et mot de passe dans le fichier de configuration de GeoNature



```
$ cd config
$ nano geonature_config.toml
PUBLIC_LOGIN = 'public'
PUBLIC_PASSWORD = 'public'
```

- Mettre à jour la configuration de GeoNature

```
$ source backend/venv/bin/activate
$ geonature update_configuration
$ sudo systemctl restart geonature
```

A ce moment là, cet utilisateur a tous les droits sur GeoNature. Il s'agit donc de gérer ses permissions dans GeoNature même.

### 3/ GeoNature

- Se connecter à GeoNature avec un utilisateur administrateur
- Aller dans le module Admin
- Cliquer sur 'Gestion des permissions'
- Choisissez l'utilisateur sélectionné
- Editer le CRUVED pour chacun des modules de l'instance. Passer à 0 tous les droits et tous les modules devant être supprimés. Laisser '3' pour les modules d'intérêt.

## 3.11 Module OCCTAX

### 3.11.1 Installer le module

Le module est fourni par défaut avec l'installation de GeoNature.

Si vous l'avez supprimé, lancez les commandes suivantes depuis le repertoire backend de GeoNature

```
source venv/bin/activate
geonature install_gn_module /home/<mon_user>/geonature/contrib/occtax occtax
```

### 3.11.2 Configuration du module

Le fichier de configuration du module se trouve ici : <GEONATURE\_DIRECTORY>/external\_modules/occtax/config/conf\_gn\_module.toml.

Pour voir l'ensemble des variables de configuration disponibles du module ainsi que leurs valeurs par défaut, ouvrir le fichier /home/<mon\_user>/geonature/external\_modules/occtax/config/conf\_gn\_module.toml.example.

Les surcouches de configuration doivent être faites dans le fichier conf\_gn\_module.toml, en ne modifiant jamais le fichier conf\_gn\_module.toml.example.

Après toute modification de la configuration d'un module, il faut régénérer le fichier de configuration du frontend comme expliqué ici : *Configuration d'un gn\_module*

## Afficher/masquer des champs du formulaire

La quasi-totalité des champs du standard Occurrences de taxons sont présents dans la base de données, et peuvent donc être saisis à partir du formulaire.

Pour plus de souplesse et afin de répondre aux besoins de chacun, l'ensemble des champs sont masquables (sauf les champs essentiels : observateur, taxon ...)

En modifiant les variables des champs ci-dessous, vous pouvez donc personnaliser le formulaire :

```
[form_fields]
    date_min = true
    date_max = true
    hour_min = true
    hour_max = true
    altitude_min = true
    altitude_max = true
    obs_technique = true
    group_type = true
    comment_releve = true
    obs_method = true
    bio_condition = true
    bio_status = true
    naturalness = true
    exist_proof = true
    observation_status = true
    diffusion_level = false
    blurring = false
    determiner = true
    determination_method = true
    sample_number_proof = true
    digital_proof = true
    non_digital_proof = true
    source_status = false
    comment_occ = true
    life_stage = true
    sex = true
    obj_count = true
    type_count = true
    count_min = true
    count_max = true
    validation_status = false
```

Si le champ est masqué, une valeur par défaut est inscrite en base (voir plus loin pour définir ces valeurs).

Par défaut le champ Observateurs est une liste déroulante qui pointe vers une liste du schéma utilisateurs. Il est possible de passer ce champ en texte libre en mettant à `true` la variable `observers_txt`.

Le paramètre `id_observers_list` permet de changer la liste d'observateurs proposée dans le formulaire. Vous pouvez modifier le numéro de liste du module ou modifier le contenu de la liste dans UsersHub (utilisateurs.`t_listes` et utilisateurs.`cor_role_liste`)

Par défaut, l'ensemble des observateurs de la liste 9 (observateurs faune/flore) sont affichés.

Le module est fourni avec une liste restreinte de taxons (8 seulement). C'est à l'administrateur de changer ou de remplir cette liste.

Le paramètre `id_taxon_list = 100` correspond à un ID de liste de la table `taxonomie.bib_listes` (L'ID 100 correspond à la liste "Saisie Occtax"). Vous pouvez changer ce paramètre avec l'ID de liste que vous souhaitez, ou bien garder cet ID et changer le contenu de cette liste.

Voici les requêtes SQL pour remplir la liste 100 avec tous les taxons de Taxref à partir du rang genre :

Il faut d'abord remplir la table `taxonomie.bib_noms` (table des taxons de sa structure), puis remplir la liste 100, avec l'ensemble des taxons de `bib_noms` :

```
DELETE FROM taxonomie.cor_nom_liste;
DELETE FROM taxonomie.bib_noms;

INSERT INTO taxonomie.bib_noms (cd_nom, cd_ref, nom_francais)
SELECT cd_nom, cd_ref, nom_vern
FROM taxonomie.taxref
WHERE id_rang NOT IN ('Dumm', 'SPRG', 'KD', 'SSRG', 'IFRG', 'PH', 'SBPH', 'IFPH', 'DV', 'SBDV',
↳ 'SPCL', 'CLAD', 'CL',
  'SBCL', 'IFCL', 'LEG', 'SPOR', 'COH', 'OR', 'SBOR', 'IFOR', 'SPFM', 'FM', 'SBFM', 'TR', 'SSTR');

INSERT INTO taxonomie.cor_nom_liste (id_liste, id_nom)
SELECT 100, n.id_nom FROM taxonomie.bib_noms n;
```

Il est également possible d'éditer des listes à partir de l'application TaxHub.

Le formulaire de saisie pré-remplit des valeurs par défaut pour simplifier la saisie. Ce sont également ces valeurs qui sont prises en compte pour remplir dans la BDD les champs du formulaire qui sont masqués.

La table `pr_occtax.defaults_nomenclatures_value` définit les valeurs par défaut pour chaque nomenclature.

La table contient les deux colonnes suivantes :

- l'`id_type` de nomenclature (voir table `ref_nomenclature.bib_nomenclatures_types`)
- l'`id_nomenclature` (voir table `ref_nomenclature.t_nomenclatures`)

Pour chaque type de nomenclature, on associe l'ID de la nomenclature que l'on souhaite voir apparaître par défaut.

Le mécanisme peut être poussé plus loin en associant une nomenclature par défaut par organisme, règne et `group2_inpn`. La valeur 0 pour ses champs revient à mettre la valeur par défaut pour tous les organismes, tous les règnes et tous les `group2_inpn`.

Une interface de gestion des nomenclatures est prévue d'être développée pour simplifier cette configuration.

TODO : valeur par défaut de la validation

La liste des champs affichés par défaut dans le tableau peut être modifiée avec le paramètre `default_maplist_columns`.

Par défaut :

```
default_maplist_columns = [
    { prop = "taxons", name = "Taxon" },
    { prop = "date_min", name = "Date début" },
    { prop = "observateurs", name = "Observateurs" },
    { prop = "dataset_name", name = "Jeu de données" }
]
```

Voir la vue `occtax.v_releve_list` pour voir les champs disponibles.

Il est possible de contraindre la saisie de la géométrie d'un relevé sur la carte par un seuil d'échelle minimum avec le paramètre `releve_map_zoom_level`.

Par défaut :

```
# Zoom level on the map from which you can add point/line/polygon
releve_map_zoom_level = 6
```

Il suffit de modifier la valeur qui correspond au niveau de zoom sur la carte. Par exemple, pour contraindre la saisie à l’affichage de la carte IGN au 1/25000e :

```
releve_map_zoom_level = 15
```

### 3.11.3 Gestion des exports

Les exports du module sont basés sur une vue (par défaut `pr_occtax.export_occtax_sinp`)

Il est possible de définir une autre vue pour avoir des exports personnalisés. Pour cela, créer votre vue, et modifier les paramètres suivants :

```
# Name of the view based export
export_view_name = 'v_export_occtax'

# Name of the geometry columns of the view
export_geom_columns_name = 'geom_4326'

# Name of the primary key column of the view
export_id_column_name = 'permId'
```

La vue doit cependant contenir les champs suivants pour que les filtres de recherche fonctionnent :

```
date_min,
date_max,
id_releve_occtax,
id_dataset,
id_occurrence_occtax,
id_digitiser,
geom_4326,
dataset_name
```

### 3.11.4 Attribuer des droits

La gestion des droits (CRUVED) se fait module par module. Cependant si on ne redéfinit pas de droit pour un module, ce sont les droits de l’application mère (GeoNature elle-même) qui seront attribués à l’utilisateur pour l’ensemble de ses sous-modules.

Pour ne pas afficher le module Occtax à un utilisateur ou à un groupe, il faut lui mettre l’action Read (R) à 0.

L’administration des droits des utilisateurs pour le module Occtax se fait dans le backoffice de gestion des permissions de GeoNature.

## 3.12 Module Admin

### 3.12.1 Administration des champs additionnels

Certains protocoles nécessitent la saisie de champs qui vont au-delà des standards du SINP sur lesquels GeoNature s’appuie. Les champs additionnels permettent ainsi d’étendre les formulaires en ajoutant des informations spécifiques pour des jeux de données (JDD) ou pour l’ensemble d’un module.

Les champs additionnels ne sont pas créés comme des colonnes à part entière, mais leurs valeurs sont stockées dans un champs `additional_data` au format JSON.

Actuellement seul le module Occtax implémente la gestion de ces champs additionnels.

Le backoffice de GeoNature offre une interface de création et de gestion de ces champs additionnels. Un champ additionnel est défini par:

- son nom (nom dans la base de données)
- son label (nom tel qu'il sera affiché sur l'interface)
- son type de widget : vous devez définir si le champs est une liste déroulante, une checkbox, une nomenclature, un entier, un champ texte, etc. . .
- le (ou les) module(s) auquel il est rattaché
- le (ou les) objet(s) auquel il est rattaché. Il s'agit du placement et de la table de rattachement du champs dans le module. Par exemple Occtax est composé de 3 "objets/table". Les objets "relevé", "occurrence" et "dénombrement".
- le (ou les) JDD auquel il est rattaché. Si aucun JDD n'est renseigné le champ sera proposé dans tout le module pour tous les JDD. S'il est rattaché à un JDD, le champs sera chargé dynamiquement à la selection du JDD dans le formulaire
- une série d'autres options pour paramétrer le comportement du champs (obligatoire, ordre, description, exportable etc. . . )

Exemples de configuration :

- Un champs type "select" :

Nom du champ *	champ_select_1
	Nom du champ en base de donnée
Label du champ *	Mon champs select n°1
	Label du champ en interface
Type Widget *	Select
Modules	× Occtax
	Module(s) auquel le champs est rattaché. *Obligatoire
Objects	× OCCTAX_RELEVÉ (Représente la table pr_occtax.t_relevés_occtax)
	Objet(s) auquel le champs est rattaché. *Obligatoire
Datasets	× ATBI de la réserve intégrale du Lauvitel dans le Parc national des Ecrins
	Jeu(x) de données auquel le champs est rattaché
Obligatoire	<input checked="" type="checkbox"/>
Description	
Quantitatif	<input type="checkbox"/>
Unité	
Valeurs	[ "Source", "Marais saumâtre", "Ruisseau/Ruisseau (< 3 m de large)", "Canal navigable" ]
	Obligatoire si widget = select/radio/bool_radio (Format JSON : tableau de valeurs ou tableau clé/valeur. Utilisez des doubles quotes pour les valeurs et les clés)
Ordre	2
	Numéro d'ordonnement du champs (si plusieurs champs pour le même module/objet/JDD)
Exportable	<input checked="" type="checkbox"/>
Type de nomenclature	
	Si Type widget = Nomenclature
Id List	
	Identifiant en BDD de la liste (pour Type widget = taxonomy/observers)
Attribut additionnels	

- Un champs type “multiselect” (la clé “value” est obligatoire dans le dictionnaire de valeurs) :



Nom du champ *	<input type="text" value="multi_select_3"/>
	Nom du champ en base de donnée
Label du champ *	<input type="text" value="Mon champs multiselect"/>
	Label du champ en interface
Type Widget *	<input type="text" value="Multiselect"/>
Modules	<input type="text" value="x Occtax"/>
	Module(s) auquel le champs est rattaché. *Obligatoire
Objects	<input type="text" value="x OCCTAX_RELEVÉ (Représente la table pr_occtax.t_relevés_occtax)"/> <input type="text" value="x OCCTAX_DENOMBREMENT (Représente la table pr_occtax.cor_counting_occtax)"/>
	Objet(s) auquel le champs est rattaché. *Obligatoire
Datasets	<input type="text" value="x ATBI de la réserve intégrale du Lauvitel dans le Parc national des Ecrins"/>
	Jeu(x) de données auquel le champs est rattaché
Obligatoire	<input type="checkbox"/>
Description	<input type="text"/>
Quantitatif	<input type="checkbox"/>
Unité	<input type="text"/>
Valeurs	<input type="text" value='[{"value": "valeur1"}, {"value": "valeur2"}, {"value": "valeur3"}]'/>
	Obligatoire si widget = select/radio/bool_radio (Format JSON : tableau de valeurs ou tableau clé/valeur. Utilisez des doubles quotes pour les valeurs et les clés)
Ordre	<input type="text" value="3"/>
	Numéro d'ordonnement du champs (si plusieurs champs pour le même module/objet/JDD)
Exportable	<input checked="" type="checkbox"/>
Type de nomenclature	<input type="text"/>
	Si Type widget = Nomenclature
Id List	<input type="text"/>
	Identifiant en BDD de la liste (pour Type widget = taxonomy/observers)
Attribut additionnels	<input type="text"/>

- Un champs type "html". C'est un champs de type "présentation", aucune valeur ne sera enregistré en base de données pour ce champs :

Label du champ *	<input type="text" value="Bloc html"/>
	Label du champ en interface
Type Widget *	<input type="text" value="Html"/>
Modules	<input type="text" value="× Occtax"/>
	Module(s) auquel le champs est rattaché. *Obligatoire
Objects	<input type="text" value="× OCCTAX_RELEVÉ (Représente la table pr_occtax.t_relevés_occtax)"/>
	Objet(s) auquel le champs est rattaché. *Obligatoire
Datasets	<input type="text"/>
	Jeu(x) de données auquel le champs est rattaché
Obligatoire	<input type="checkbox"/>
Description	<input type="text" value="Un champs HTML. C'est un simple champs d'affichage, rien ne sera stocké en base de données"/>
Quantitatif	<input type="checkbox"/>
Unité	<input type="text"/>
Valeurs	<input type="text"/>
	Obligatoire si widget = select/radio/bool_radio (Format JSON : tableau de valeurs ou tableau clé/valeur. Utilisez des doubles quotes pour les valeurs et les clés)
Ordre	<input type="text" value="1"/>
	Numéro d'ordonnement du champs (si plusieurs champs pour le même module/objet/JDD)
Exportable	<input checked="" type="checkbox"/>
Type de nomenclature	<input type="text"/>
	Si Type widget = Nomenclature
Id List	<input type="text"/>
	Identifiant en BDD de la liste (pour Type widget = taxonomy/observers)
Attribut additionnels	<input type="text" value='{"html": "Ce texte sera affiché dans le bloc html", "class": "alert alert-primary"}'/>

## Configuration avancée des champs

Le champs "Attribut additionnels" permet d'ajouter des éléments de configuration sur les formulaires sous forme de JSON: - Ajouter une icône "?" et un tooltip au survol du formulaire : `{"description": "mon tooltip"}` - Ajouter un sous-titre descriptif : `{"help": "mon sous titre"}` - Ajouter des valeurs min/max pour un input number : `{"min": 1, "max": 10}`

## 3.13 Module OCCHAB

### 3.13.1 Installer le module

Le module OCCHAB fait parti du coeur de GeoNature. Son installation est au choix de l'administrateur.

Pour l'installer, lancer les commande suivante:

```
cd backend
source venv/bin/activate
geonature install_gn_module /home/`whoami`/geonature/contrib/gn_module_occhab occtax
```

### 3.13.2 Base de données

Le module s'appuie sur deux schémas.: `ref_habitats`: Le référentiel habitat du MNHN `pr_occhab`: le schéma qui contient les données d'occurrence d'habitat, basé sur standard du MNHN

### 3.13.3 Configuration

Le parametrage du module OCCHAB se fait depuis le fichier `/home/`whoami`/geonature/contrib/config/conf_gn_module.toml` Après toute modification de la configuration d'un module, il faut régénérer le fichier de configuration du frontend comme expliqué ici : *Configuration d'un gn\_module*

#### Formulaire

- La liste des habitats fournit pour la saisie est basé sur une liste définit en base de données (table `ref_habitat.cor_list_habitat` et `ref_habitat.bib_list_habitat`). Il est possible d'éditer cette liste directement den base de données, d'en créer une autre et de changer la liste utiliser par le module. Editer alors ce paramètre:

```
ID_LIST_HABITAT = 1
```

- Le formulaire permet de saisir des observateur basés sur le referentiel utilisateurs ( `false`) ou de les saisir en texte libre (`true`).

```
OBSERVER_AS_TXT = false
```

- L'ensemble des champs du formulaire son masquables. Pour en masquer certains, passer à `false` les variables suivantes:

```
[formConfig]
date_min = true
date_max = true
depth_min = true
depth_max = true
altitude_min = true
altitude_max = true
exposure = true
area = true
comment = true
area_surface_calculation = true
geographic_object = true
determination_type = true
determiner = true
```

(continues on next page)

(continued from previous page)

```
collection_technique = true
technical_precision = true
recovery_percentage = true
abundance = true
community_interest = true
```

Voir le fichier `conf_gn_module.toml.example` qui liste l'ensemble des paramètres de configuration du module.

## 3.14 Module SYNTHÈSE

Le module Synthèse est un module du coeur de GeoNature, fourni par défaut lors de l'installation.

### 3.14.1 Configuration

L'ensemble des paramètres de configuration du module se trouve dans le fichier général de configuration de GeoNature `config/geonature_config.toml` puisqu'il s'agit d'un module du coeur.

#### 1. Modifier les filtres géographiques disponibles par défaut dans l'interface de recherche.

Editer la variable `AREA_FILTERS` en y ajoutant le label et le code du type d'entité géographique que vous souhaitez rajouter. Voir table `ref_geo.bib_areas_types`. Dans l'exemple on ajoute le type `ZNIEFF1` (`code_type = "ZNIEFF1"`). Attention, dans ce cas les entités géographiques correspondantes au type `ZNIEFF1`, doivent également être présentes dans la table `ref_geo.l_areas`. Attention : Si des données sont déjà présentes dans la synthèse et que l'on ajoute de nouvelles entités géographiques à `ref_geo.l_areas`, il faut également recalculer les valeurs de la table `gn_synthese.cor_area_synthese` qui assure la correspondance entre les données de la synthèse et les entités géographiques.

```
[SYNTHÈSE]
# Liste des entités géographiques sur lesquels les filtres
# géographiques de la synthèse s'appuient (type_code = code du type de l'entité_
→géo, table ref_geo.bib_areas_types)
AREA_FILTERS = [
  { label = "Communes", "type_code": "COM" },
  { label = "ZNIEFF1", "type_code": "ZNIEFF1" },
]
```

Il est aussi possible de passer plusieurs `type_code` regroupés dans un même filtre géographique (exemple : { label = "Zonages réglementaires", type\_code = ["ZC", "ZPS", "SIC"] }).

#### 2. Configurer les champs des exports

Dans tous les exports, l'ordre et le nom des colonnes sont basés sur la vue servant l'export. Il est possible de les modifier en éditant le SQL des vues en respectant bien les consignes ci-dessous.

##### Export des observations

Les exports (CSV, GeoJson, Shapefile) sont basés sur la vue `gn_synthese.v_synthese_for_export`.

Il est possible de ne pas intégrer certains champs présents dans cette vue d'export. Pour cela modifier le paramètre `EXPORT_COLUMNS`.

Enlevez la ligne de la colonne que vous souhaitez désactiver. Les noms de colonne de plus de 10 caractères seront tronqués dans le fichier shapefile.

```
[SYNTHESE]
EXPORT_COLUMNS = [
    "date_debut",
    "date_fin",
    "heure_debut",
    "heure_fin",
    "cd_nom",
    "cd_ref",
    "nom_valide",
    "nom_vernaculaire",
    "nom_cite",
    "regne",
    "group1_inpn",
    "group2_inpn",
    "classe",
    "ordre",
    "famille",
    "rang_taxo",
    "nombre_min",
    "nombre_max",
    "alti_min",
    "alti_max",
    "prof_min",
    "prof_max",
    "observateurs",
    "determineur",
    "communes",
    "x_centroid_4326",
    "y_centroid_4326",
    "geometrie_wkt_4326",
    "nom_lieu",
    "comment_releve",
    "comment_occurrence",
    "validateur",
    "niveau_validation",
    "date_validation",
    "comment_validation",
    "preuve_numerique_url",
    "preuve_non_numerique",
    "jdd_nom",
    "jdd_uuid",
    "jdd_id",
    "ca_nom",
    "ca_uuid",
    "ca_id",
    "cd_habref",
    "cd_habitat",
    "nom_habitat",
    "precision_geographique",
    "nature_objet_geo",
    "type_regroupement",
    "methode_regroupement",
    "technique_observation",
    "biologique_statut",
    "etat_biologique",
    "biogeographique_statut",
    "naturalite",
```

(continues on next page)

(continued from previous page)

```

    "preuve_existante",
    "niveau_precision_diffusion",
    "stade_vie",
    "sexe",
    "objet_denombrement",
    "type_denombrement",
    "niveau_sensibilite",
    "statut_observation",
    "floutage_dee",
    "statut_source",
    "type_info_geo",
    "methode_determination",
    "comportement",
    "reference_biblio",
    "id_synthese",
    "id_origine",
    "uuid_perm_sinp",
    "uuid_perm_grp_sinp",
    "date_creation",
    "date_modification"
]

```

**Note** L'entête [SYNTHESE] au dessus EXPORT\_COLUMNS indique simplement que cette variable appartient au bloc de configuration de la synthèse. Ne pas rajouter l'entête à chaque paramètre de la synthèse mais une seule fois au dessus de toutes les variables de configuration du module.

Il est également possible de personnaliser ses exports en éditant le SQL de la vue `gn_synthese.v_synthese_for_export` (niveau SQL et administration GeoNature avancé).

Attention, certains champs sont cependant obligatoires pour assurer la réalisation des fichiers d'export (csv, geojson et shapefile) et des filtres CRUVED.

La vue doit OBLIGATOIREMENT contenir les champs :

- `geojson_4326`
- `geojson_local`
- `id_synthese`,
- `jdd_id` (l'ID du jeu de données)
- `id_digitiser`
- `observateurs`

Ces champs doivent impérativement être présents dans la vue, mais ne seront pas nécessairement dans le fichier d'export si ils ne figurent pas dans la variable `EXPORT_COLUMNS`. De manière générale, préférez rajouter des champs plutôt que d'en enlever !

Le nom de ces champs peut cependant être modifié. Dans ce cas, modifiez le fichier `geonature_config.toml`, section `SYNTHESE` parmi les variables suivantes (`EXPORT_ID_SYNTHESE_COL`, `EXPORT_ID_DATASET_COL`, `EXPORT_ID_DIGITISER_COL`, `EXPORT_OBSERVERS_COL`, `EXPORT_GEOJSON_4326_COL`, `EXPORT_GEOJSON_LOCAL_COL`).

NB : Lorsqu'on effectue une recherche dans la synthèse, on interroge la vue `gn_synthese.v_synthese_for_web_app`. L'interface web passe ensuite une liste d'`id_synthese` à la vue `gn_synthese.v_synthese_for_export` correspondant à la recherche précédemment effectuée (ce qui permet à cette seconde vue d'être totalement modifiable).

La vue `gn_synthese.v_synthese_for_web_app` est taillée pour l'interface web, il ne faut donc PAS la modifier.

### Export des métadonnées

En plus des observations brutes, il est possible d'effectuer un export des métadonnées associées aux observations. L'export est au format CSV et est construit à partir de la table `gn_synthese.v_metadata_for_export`. Vous pouvez modifier le SQL de création de cette vue pour customiser votre export (niveau SQL avancé).

Deux champs sont cependant obligatoire dans la vue :

- `jdd_id` (qui correspond à l'id du JDD de la table `gn_meta.t_datasets`). Le nom de ce champs est modifiable. Si vous le modifiez, éditez la variable `EXPORT_METADATA_ID_DATASET_COL`.
- `acteurs`: Le nom de ce champs est modifiable. Si vous le modifiez, éditez la variable `EXPORT_METADATA_ACTOR_COL`

### Export des statuts taxonomiques (réglementations)

Cet export n'est pas basé sur une vue, il n'est donc pas possible de l'adapter.

### 3. Configurer les seuils du nombre de données pour la recherche et les exports

Par défaut et pour des questions de performance (du navigateur et du serveur) on limite à 50000 le nombre de résultat affiché sur la carte et le nombre d'observations dans les exports.

Ces seuils sont modifiables respectivement par les variables `NB_MAX_OBS_MAP` et `NB_MAX_OBS_EXPORT` :

Le mode cluster activé par défaut peut être désactivé via le paramètre `ENABLE_LEAFLET_CLUSTER`. Dans ce cas, il est conseillé de repasser le paramètre `NB_MAX_OBS_MAP` à 10000.

```
[SYNTHESE]
# Nombre d'observation maximum à afficher sur la carte après une recherche
NB_MAX_OBS_MAP = 10000
# Nombre max d'observation dans les exports
NB_MAX_OBS_EXPORT = 40000
```

### 4. Désactiver des filtres génériques

L'interface de recherche de la synthèse permet de filtrer sur l'ensemble des nomenclatures de la table `gn_synthese`, il est cependant possible de désactiver les filtres de certains champs.

Modifiez la variable `EXCLUDED_COLUMNS`

```
[SYNTHESE]
EXCLUDED_COLUMNS = ['non_digital_proof'] # pour enlever le filtre 'preuve non_
↪numérique'
```

D'autres éléments sont paramétrables dans le module `synthese`. La liste complète est disponible dans le fichier `config/geonature_config.toml` rubrique `SYNTHESE`.



## 3.15 Module VALIDATION

Le module VALIDATION, intégré depuis la version 2.1.0 dans le cœur de GeoNature permet de valider des occurrences de taxon en s'appuyant sur les données présentes dans la SYNTHÈSE. Le module s'appuie sur le [standard Validation](#) du SINP et sur ses [nomenclatures officiels](#).

Afin de valider une occurrence, celle-ci doit impérativement avoir un UUID. En effet, la validation est stockée en BDD dans la table transversale `gn_commons.t_validations` ([voir doc](#)) qui impose la présence de cet UUID.

La table `gn_commons.t_validations` contient l'ensemble de l'historique de validation des occurrences. Pour une même occurrence (identifiée par un UUID unique) on peut donc retrouver plusieurs lignes dans la table correspondant au différents statuts de validation attribués à cet occurrence dans le temps.

La vue `gn_commons.v_latest_validation` permet de récupérer le dernier statut de validation d'une occurrence.

NB : une donnée non présente dans la SYNTHÈSE, ne remontera pas dans l'interface du module VALIDATION. Cependant rien n'empêche un administrateur avancé d'utiliser la table de validation et son mécanisme pour des données qui ne seraient pas en SYNTHÈSE (du moment que les données disposent d'un UUID).

Au niveau de l'interface, le formulaire de recherche est commun avec le module SYNTHÈSE. Les paramètres de configuration du formulaire sont donc également partagés et administrables depuis le fichier `geonature_config.toml`, rubrique SYNTHÈSE.

### 3.15.1 Configuration

Le paramétrage du module VALIDATION se fait depuis le fichier `/home/~whoami~/geonature/contrib/gn_module_validation/config/conf_gn_module.toml` Après toute modification de la configuration d'un module, il faut régénérer le fichier de configuration du frontend comme expliqué ici : *Configuration d'un gn\_module*

#### Liste des champs visibles

Gestion de l'affichage des colonnes de la liste via le paramètre `COLUMN_LIST` :

```
[[COLUMN_LIST]]
column_label = "nomenclature_life_stage.label_default" # Champs de la synthèse, ↵
↳ éventuellement en suivant des relationships
column_name = "Stade de vie" # Titre de la colonne
min_width = 100 # Taille minimale de la colonne
max_width = 100 # Taille maximale de la colonne
```

#### E-mail

Il est possible de personnaliser le message du mail envoyé aux observateurs. Pour ce faire il faut modifier les paramètres `MAIL_BODY` et `MAIL_SUBJECT`

Pour afficher dans le mail des données relatives à l'observation ou au taxon il faut respecter la syntaxe suivante: `${d.NOM_PROPRIETE }`

Liste des propriétés disponibles :

- communes : liste des communes
- medias : Titre, auteur et lien vers le média associée

- `data_link` : lien vers l'observation dans son module de saisie
- tous les champs de la synthèse (`acquisition_framework`, `altitude_max`, `altitude_min`, `bio_status`, `blurring`, `cd_hab`, `cd_nom`, `comment_context`, `comment_description`, `date_min`, `depth_max`, `depth_min`, `determiner`, `diffusion_level`, `digital_proof`, `entity_source_pk_value`, `exist_proof`, `grp_method`, `grp_typ`, `last_action`, `life_stage`, `meta_create_date`, `meta_update_date`, `meta_v_taxref`, `meta_validation_date`, `nat_obj_geo`, `naturalness`, `nom_cite`, `non_digital_proof`, `obj_count`, `obs_technique`, `observation_status`, `observers`, `occ_behaviour`, `occ_stat_biogeo`, `place_name`, `precision`, `sample_number_proof`, `sensitivity`, `sex`, `source`, `type_count`, `unique_id_sinp`, `unique_id_sinp_grp`, `valid_status`, `validation_comment`)
- tous les champs du taxon (`cd_nom`, `cd_ref`, `cd_sup`, `cd_taxsup`, `regne`, `ordre`, `classe`, `famille`, `group1_inpn`, `group2_inpn`, `id_rang`, `nom_complet`, `nom_habitat`, `nom_rang`, `nom_statut`, `nom_valide`, `nom_vern`)



## IMPORT NIVEAU 1

Dans cet exemple, nous allons importer un fichier CSV (ou SHP) d'observations dans la base de données de GeoNature, pour ensuite intégrer ces données dans la synthèse de GeoNature.

On utilisera le fichier d'exemple <https://github.com/PnX-SI/Ressources-techniques/blob/master/GeoNature/V2/import-basique/01-observations-faune-2008-2010.csv>.

### 4.1 Importer la donnée source dans la BDD avec QGIS

PS : Si vous utilisez un CSV, vous pouvez aussi utiliser la fonction `gn_imports.load_csv_file`.

#### 1. Connecter la BDD dans QGIS :

- QGIS
- PostGIS / Clic droit / New connection
- Nom / Hôte (IP) / Base de données (geonaturedb) / Authentification de base (utilisateur / mot de passe)
- Parcourir les tables géométriques

Si vous devez ouvrir les connexions externes à votre BDD, voir la documentation <https://github.com/PnEcrins/GeoNature-atlas/blob/master/docs/installation.rst#acc%C3%A9der-%C3%A0-votre-bdd>

#### 2. Importer le fichier dans la BDD :

- Ouvrir SHP ou CSV dans QGIS
- Bases de données / Gestionnaire de base de données
- Sélectionner la BDD et son schéma
- Importer une couche/un fichier
- Choisir la couche à importer et définir le nom de table de destination

### 4.2 Créer les métadonnées

#### 1. Ajouter une source (si elle n'existe pas déjà)

En l'ajoutant manuellement dans la table `gn_synthese.t_sources` ou en SQL :

```
INSERT INTO gn_synthese.t_sources(name_source, desc_source)
VALUES
('Historique', 'Données historiques intégrées manuellement dans la BDD')
```

## 2. Ajouter un jeu de données (si il n'existe pas déjà)

Avec l'admin de GeoNature, dans la BDD avec pgAdmin ou en SQL. Et avant ça un CA si il n'en existe pas déjà un auquel associer le JDD.

```
INSERT INTO gn_meta.t_datasets(id_acquisition_framework, dataset_name, dataset_
↳shortname, dataset_desc, id_nomenclature_data_type, keywords, marine_domain,↳
↳terrestrial_domain, active)
VALUES
(1, 'Données Faune 2008-2010', 'Faune 2008-2010', 'Données faune du PNE entre 2008 et_
↳2010', 326, 'Faune, PNE', FALSE, TRUE, TRUE)
```

Pour retrouver les valeurs d'un type de nomenclature, vous pouvez utiliser les vues qui les rendent plus lisibles. Par exemple ici `ref_nomenclatures.v_data_typ`. Ou bien l'Admin des nomenclatures disponible dans GeoNature.

Il est aussi possible d'utiliser les codes des nomenclatures pour retrouver leurs id (ceci étant variables d'une instance à l'autre), en utilisant la fonction `ref_nomenclature.get_id_nomenclature`.

## 4.3 Insertion des données dans la Synthèse

```
INSERT INTO gn_synthese.synthese(
unique_id_sinp,
id_source,
id_dataset,
id_nomenclature_obs_meth,
count_min,
count_max,
cd_nom,
nom_cite,
altitude_min,
altitude_max,
the_geom_4326,
the_geom_point,
the_geom_local,
date_min,
date_max,
observers,
comments,
last_action
)
SELECT
    uuid_generate_v4(), -- Attention, ne générez un UUID_SINP pour chaque obs que_
↳si vous êtes surs qu'elles n'en ont pas déjà un
    2 AS id_source,
    3 AS id_dataset,
    CASE
        WHEN critere = 'Vu' THEN (41) -- Ou bien ref_nomenclature.get_id_nomenclature
        WHEN critere = 'Entendu' THEN (42)
        ELSE (gn_synthese.get_default_nomenclature_value('METH_OBS'))
    END AS id_nomenclature_obs_meth,
    effectif::integer,
    effectif::integer,
    cd_nom::integer,
    taxon_latin,
    altitude::integer, -- On pourrait calculer les valeurs manquantes avec la_
↳fonction ref_geo.fct_get_altitude_intersection
```

(continues on next page)

(continued from previous page)

```

        altitude::integer,
        ST_SetSRID(ST_MakePoint("x_WGS84"::numeric, "y_WGS84"::numeric),4326) AS the_
↪geom_4326,
        ST_Centroid(ST_SetSRID(ST_MakePoint("x_WGS84"::numeric, "y_WGS84"::numeric),
↪4326)) AS the_geom_point,
        ST_Transform(ST_SetSRID(ST_MakePoint("x_WGS84"::numeric, "y_WGS84"::numeric),
↪4326),2154) AS the_geom_local,
        dateobs::date,
        dateobs::date,
        observateurs,
        remarques,
        'I' AS last_action -- code de la dernière action effectuée: Valeurs possibles
↪'I': insert, 'U': update
FROM gn_imports.obs_faune_2008_2010
ORDER BY dateobs
;

```

A creuser pour calculer les altitudes non renseignées :

```

SELECT id_synthese,
(ref_geo.fct_get_altitude_intersection(the_geom_local)).altitude_min
(ref_geo.fct_get_altitude_intersection(the_geom_local)).altitude_max
FROM gn_synthese.synthese
LIMIT 1000;

```

Gil propose de rajouter une PK et de faire un lien entre les données de la table importée et celles dans la synthèse avec `entity_source_pk_value` :

```

-- Clé primaire
ALTER TABLE gn_imports.obs_faune_2008_2010
    ADD COLUMN gid serial;

ALTER TABLE gn_imports.obs_faune_2008_2010
    ADD CONSTRAINT pk_obs_faune_2008_2010 PRIMARY KEY (gid);

```

Ajouter le champ `entity_source_pk_value` dans ton INSERT et `gid` dans le SELECT.

On pourrait aussi remplir `cor_observers_synthese` si on le veut et si on a les observateurs présents dans les données, en les faisant correspondre avec leurs `id_role`.

L'intégration de données dans la Synthèse peut faire apparaître des nouveaux taxons présents sur le territoire. Si vous souhaitez les proposer à la saisie dans Occtax, il faut les ajouter dans `taxonomie.bib_noms` puis dans la liste "Saisie Occtax".

```

-- Remplir taxonomie.bib_noms avec les nouveaux noms présents dans la synthèse
INSERT INTO taxonomie.bib_noms (cd_nom, cd_ref)
SELECT DISTINCT s.cd_nom, t.cd_ref
FROM gn_synthese.synthese s
JOIN taxonomie.taxref t
ON s.cd_nom = t.cd_nom
WHERE not s.cd_nom IN (SELECT DISTINCT cd_nom FROM taxonomie.bib_noms);

```

Il faudrait ensuite les ajouter à la liste "Saisie Occtax", pour que ces nouveaux noms soient proposés à la saisie dans le module Occtax de GeoNature.

L'installation de GeoNature intègre les communes de toute la France métropolitaine. Pour alléger la table `ref_geo.l_areas`, il peut être pertinent de supprimer les communes en dehors du territoire de travail. Par exemple, supprimer toutes les communes en dehors du département.

Pour retrouver le détail de toutes les communes du département Bouches-du-Rhône :

```
SELECT * FROM ref_geo.l_areas la
JOIN ref_geo.bib_areas_types ba ON ba.id_type = la.id_type
JOIN ref_geo.li_municipalities lm ON lm.id_area = la.id_area
WHERE ba.type_code = 'COM' AND lm.insee_dep = '13'
```

A utiliser dans une requête de suppression, en gérant les cascades entre les tables.

## 4.4 Insertion depuis un shapefile

L'exercice est similaire si on part depuis un fichier Shape (<https://github.com/PnX-SI/Ressources-techniques/blob/master/GeoNature/V2/import-basique/01-observations-faune-2008-2010-SHP.zip>)

La seule différence est que la géométrie est calculée lors de l'import de QGIS vers PostGIS.

Ainsi la partie Géométrie de la requête d'insertion dans la Synthèse serait :

```
ST_Transform(ST_SetSRID(geom,2154),4326 AS the_geom_4326,
ST_Centroid(ST_SetSRID(geom,2154) AS the_geom_point,
geom AS the_geom_local,
```



## IMPORT NIVEAU 2

### 5.1 Description

L'exercice consiste à importer le fichier <https://github.com/PnX-SI/Ressources-techniques/blob/master/GeoNature/V2/import-avance/observations.csv> dans GeoNature V2.

- 1 - On charge le fichier CSV dans une table de la base de données.
- 2 - On prépare la table importée (FK et typage des champs si besoin).
- 3 - On crée les métadonnées pour que GeoNature sache identifier les nouvelles données.
- 4 - On mappe les champs de la table d'import avec ceux de la synthèse.

Pour cela on utilise une fonction dédiée qui nous prépare le travail. Il ne reste plus qu'à finaliser le mapping (la fonction ne peut pas tout deviner).

- 5 - On crée la requête d'import.

Pour cela on utilise une fonction dédiée qui nous prépare le travail. On adapte la requête produite par la fonction.

- 6 - On importe les données en synthèse.
- 7 - On gère les nouveaux taxons vis à vis la saisie.
- 8 - On archive la table où on veut.

### 5.2 1 - Import du fichier CSV

Cette action doit être faite par un superuser PostgreSQL.

#### notes

- Le fichier CSV doit être présent localement sur le serveur hébergeant la base de données.
- Il fichier doit être encodé en UTF-8 et la première ligne doit comporter le nom des champs.
- Le séparateur de champs doit être le point-virgule.
- La fonction utilise la fonction `COPY` capable de lire le système de fichier du serveur. Pour des raisons de sécurité, cette fonction `COPY` n'est accessible qu'aux superutilisateurs. Vous devez donc disposer d'un accès superutilisateur PostgreSQL pour utiliser cette fonction d'import. Si l'utilisateur connecté à la base dans pgAdmin n'est pas superuser, on peut le faire dans psql.

DANS UN TERMINAL :

```
sudo su postgres
psql -d geonature2db
```

Dans les 2 cas, copier-coller les 2 commandes ci-dessous en adaptant les chemins et l'utilisateur.

```
SELECT gn_imports.load_csv_file('/home/myuser/imports/observations.csv', 'gn_imports.
↳testimport');
ALTER TABLE gn_imports.testimport OWNER TO geonatuser;
```

#### notes

- Attention : si la table existe, elle est supprimée et recrée à partir du CSV fourni.
- La fonction crée la table et sa structure dans le schéma et la table fournie en paramètre.
- Le contenu du fichier CSV est chargé dans la table (initialement toutes les colonnes sont de type text).
- La fonction tente ensuite d'identifier et de modifier le type de chacune des colonnes à partir du contenu et seuls les types integer, real, et date sont actuellement reconnus.
- Si vous devez modifier manuellement le type d'une colonne, vous pouvez vous inspirer du code ci-dessous.

## 5.3 2 - Préparation de la nouvelle table contenant les données importées

### Ajouter une clé primaire sur la table importée

```
ALTER TABLE gn_imports.testimport ADD PRIMARY KEY (id_data);
```

Si la table ne comporte pas de champ permettant une identification unique des observations :

```
ALTER TABLE gn_imports.testimport
  ADD COLUMN gid serial;

ALTER TABLE gn_imports.testimport
  ADD CONSTRAINT pk_testimport PRIMARY KEY (gid);
```

### Si besoin de mettre à jour le type de certains champs de la table importée

```
ALTER TABLE monschema.matable ALTER COLUMN macolonne TYPE montype USING
↳macolonne::montype;
```

En l'état vos données sont importées et stockées dans la base GeoNature. Cependant GeoNature ne connaît pas ces données. Pour qu'elles soient utilisables, au moins en consultation, vous devez fournir à l'application GeoNature un certain nombre d'informations concernant ces données et à minima les importer dans la synthèse. Vous pouvez également les importer dans un autre module, comme "Occtax" (non abordé dans cet exemple).

## 5.4 3 - Création des métadonnées

Il est nécessaire de rattacher les données importées à un jeu de données qui doit appartenir à un cadre d'acquisition. Si ceux-ci n'ont pas encore été créés dans la base, vous devez le faire dans `gn_meta.t_acquisition_frameworks` pour le cadre d'acquisition et dans `gn_meta.t_datasets` pour le jeu de données. Vous pouvez pour cela utiliser les formulaires disponibles dans l'interface d'administration de GeoNature : <http://myurl/geonature/#/admin>

Le jeu de données doit être rattaché à un protocole décrivant la manière dont les données ont été collectées.

**notes** Noter les ID retournés lors des insertions.

```
INSERT INTO gn_meta.sinp_datatype_protocols ( protocol_name, protocol_desc, id_
↳nomenclature_protocol_type, protocol_url)
VALUES ('ATBI', 'Inventaire ATBI Dans la réserve de Lauvitel - Date_debut : 2013-01-
↳01 - Date_fin : 3000-01-01', 395, NULL) returning id_protocol;
```

```
INSERT INTO gn_meta.t_datasets(id_acquisition_framework, dataset_name, dataset_
↳shortname, dataset_desc, id_nomenclature_data_type, keywords, marine_domain,
↳terrestrial_domain, active)
VALUES (1, 'Observations Flavia 2017', 'Observations Flavia 2017', 'Données ATBI_
↳Flavia pour l''année 2017', 326, 'Invertébrés, PNE, ATBI', FALSE, TRUE, TRUE)
↳returning id_dataset;
```

Il est également nécessaire, pour la synthèse, de lui indiquer où sont stockées les données qu'elle contient et comment y accéder. Vous devez pour cela disposer d'une source de données dans `gn_synthese.t_sources` correspondant aux données à importer. Pour l'exemple nous allons créer une source de données avec la commande SQL suivante :

```
INSERT INTO gn_synthese.t_sources(name_source, desc_source)
VALUES('ATBI', 'Données d''inventaire ATBI') returning id_source;
```

**notes**

- D'autres valeurs sont attendues mais pour l'exercice, le fichier source utilise des valeurs insérée à titre d'exemple lors de la création de la base GeoNature.
- `id_role` 3 et 4 dans `utilisateurs.t_roles`
- `id_organisme` 1 dans `utilisateurs.bib_organismes`

**notes** Il est possible d'utiliser ce mécanisme générique pour insérer des données de n'importe quelle table vers n'importe quelle autre, à partir du moment où il est possible d'établir un mapping cohérent entre les champs et notamment que les types puissent correspondre ou soient "transtypables".

## 5.5 4 - Création du mapping (source → cible)

Le schéma `gn_imports` comporte trois tables permettant de préparer le mapping des champs entre la table importée (source) et une table de destination (target).

- `gn_imports.matching_tables` permet de déclarer la table source et la table de destination. Noter le `id_matching_table` généré par la séquence lors de l'insertion d'un nouveau "matching" dans cette table.
- `gn_imports.matching_fields` permet de faire le matching entre les champs de la table source et de la table de destination. Vous devez indiquer le type de chacun des champs de la table de destination ainsi que le `id_matching_table`.

- `gn_imports.matching_geoms` permet de préparer la création du geom dans la table de destination à partir du ou des champs constituant le geom fourni dans la table source : champs contenant les x et y pour un format xy ou le champ comportant le wkt pour le format wkt.

En attendant la création d'une interface permettant de faciliter l'import, vous devez remplir ces tables manuellement. Cependant, la fonction `gn_imports.fct_generate_mapping('table_source', 'table_cible', forcedelete)` permet de pré-générer un mapping.

Si le mapping source/cible existe, la fonction ne fait rien et un message d'erreur est levé. Si le mapping n'existe pas ou si le paramètre `forcedelete` (boolean default = false) est à true, la fonction crée le mapping en remplissant la table `gn_imports.matching_tables` et la table `gn_imports.matching_fields` avec une ligne par champ de la table cible. Il ne vous reste plus qu'à manuellement supprimer ou remplacer les valeurs 'replace me' dans le champs `source_field` ou les valeurs par défaut proposées par la fonction.

### Pré-générer les champs à mapper

```
SELECT gn_imports.fct_generate_matching('gn_imports.testimport', 'gn_synthese.synthese'
↪);
SELECT gn_imports.fct_generate_matching('gn_imports.testimport', 'gn_synthese.cor_
↪observer_synthese');
```

OU si besoin d'écraser un mapping des champs existants

```
SELECT gn_imports.fct_generate_matching('gn_imports.testimport', 'gn_synthese.synthese'
↪, true);
SELECT gn_imports.fct_generate_matching('gn_imports.testimport', 'gn_synthese.cor_
↪observer_synthese', true);
```

IL FAUT ICI METTRE A JOUR LA TABLE `gn_imports_matching_fields` pour établir manuellement la correspondance des champs entre la table source et la table cible (voir le mapping final pour le fichier CSV fourni en exemple à la fin de cette page).

#### notes

- Au moins un des 2 champs `source_field` ou `source_default_value` doit être renseigné.
- Si le champ `source_field` est renseigné, le champ `source_default_value` est ignoré.

Une fois que le mapping est renseigné, vous pouvez passer à l'étape suivante.

## 5.6 5 - Construire la requête d'import

Attention, pgAdmin va tronquer le résultat. Pour obtenir l'ensemble de la requête utiliser le bouton d'export du résultat dans un fichier ou exécuté la requête avec psql.

### Génération de la requête d'import dans les tables de destination

```
SELECT gn_imports.fct_generate_import_query('gn_imports.testimport', 'gn_synthese.
↪synthese');
SELECT gn_imports.fct_generate_import_query('gn_imports.testimport', 'gn_synthese.cor_
↪observer_synthese');
```

**notes** UTILISER LE BOUTON D'EXPORT DU RESULTAT DE LA REQUETE DE PGADMIN ou utiliser psql. IL EST NECESSAIRE D'ADAPTER LA REQUETE SI BESOIN DE FAIRE DES JOIN POUR RECUPERER DES VALEURS DANS D'AUTRES TABLES

## 5.7 6 - Chargement des données dans la table de destination (synthese ici)

Voir la requête d'import en synthèse à la fin de cette page.

## 5.8 7 - On gère les nouveaux taxons vis à vis la saisie

Gestion des taxons dans `taxonomie.bib_noms` et de la liste des taxons saisissables dans `Occtax`.

Cette étape est optionnelle et va permettre de rajouter les nouveaux taxons intégrés dans la synthèse dans la table des taxons de votre territoire (`taxonomie.bib_noms`) et dans la liste des taxons saisissables dans `Occtax` (`cor_nom_liste`).

### Création d'une table temporaire

```
CREATE TABLE gn_imports.new_noms
(
  cd_nom integer NOT NULL,
  cd_ref integer NOT NULL,
  nom_fr character varying,
  array_listes integer[],
  CONSTRAINT new_noms_pkey PRIMARY KEY (cd_nom)
);
```

### Insertion des nouveaux taxons dans cette table et calcul des listes

```
TRUNCATE TABLE gn_imports.new_noms;
INSERT INTO gn_imports.new_noms
SELECT DISTINCT
  i.cd_nom,
  t.cd_ref,
  split_part(t.nom_vern, ',', 1),
  array_agg(DISTINCT l.id_liste) AS array_listes
FROM gn_imports.testimport i
LEFT JOIN taxonomie.taxref t ON t.cd_nom = i.cd_nom
LEFT JOIN taxonomie.bib_listes l ON id_liste = 100
WHERE i.cd_nom NOT IN (SELECT cd_nom FROM taxonomie.bib_noms)
GROUP BY i.cd_nom, t.cd_ref, nom_vern;
```

### Insertion dans ``bib\_noms``

```
SELECT setval('taxonomie.bib_noms_id_nom_seq', (SELECT max(id_nom) FROM taxonomie.bib_noms), true);
INSERT INTO taxonomie.bib_noms(cd_nom, cd_ref, nom_francais)
SELECT cd_nom, cd_ref, nom_fr FROM gn_imports.new_noms;
```

### Insertion dans ``cor\_nom\_liste``

```
INSERT INTO taxonomie.cor_nom_liste (id_liste, id_nom)
SELECT unnest(array_listes) AS id_liste, n.id_nom
FROM gn_imports.new_noms tnn
JOIN taxonomie.bib_noms n ON n.cd_nom = tnn.cd_nom;
```

Si on veut nettoyer et qu'on est sûr de ne plus en avoir besoin

```
DROP TABLE gn_imports.new_noms;
```

## 5.9 8 - Déplacement de la table importée (facultatif)

On peut si on le souhaite déplacer la table vers une destination d'archivage

```
ALTER TABLE gn_imports.testimport SET SCHEMA schema_destination;
```

On peut la mettre dans le schéma gn\_exports pour l'exercice afin de tester mais ce n'est pas sa vocation.

## 5.10 RESULTAT FINAL

```
--DELETE FROM gn_imports.matching_fields WHERE id_matching_table IN (1,2);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (207, NULL, 'uuid_generate_v4()', 'unique_id_sinp', 'uuid', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (208, NULL, 'uuid_generate_v4()', 'unique_id_sinp_grp', 'uuid', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (219, NULL, 'gn_synthese.get_default_nomenclature_value(''PREUVE_EXIST'
↳'::character varying)', 'id_nomenclature_exist_proof', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (210, 'id_data', NULL, 'entity_source_pk_value', 'character varying', NULL,
↳1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (211, 'id_lot', NULL, 'id_dataset', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (209, 'id_source', NULL, 'id_source', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (213, NULL, 'gn_synthese.get_default_nomenclature_value(''TYP_GRP'
↳'::character varying)', 'id_nomenclature_grp_typ', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (212, NULL, 'gn_synthese.get_default_nomenclature_value(''NAT_OBJ_GEO'
↳'::character varying)', 'id_nomenclature_geo_object_nature', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (214, NULL, 'gn_synthese.get_default_nomenclature_value(''METH_OBS'
↳'::character varying)', 'id_nomenclature_obs_meth', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (215, NULL, 'gn_synthese.get_default_nomenclature_value(''TECHNIQUE_OBS'
↳'::character varying)', 'id_nomenclature_obs_technique', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (217, NULL, 'gn_synthese.get_default_nomenclature_value(''ETA_BIO'
↳'::character varying)', 'id_nomenclature_bio_condition', 'integer', NULL, 1);
```

(continues on next page)

(continued from previous page)

```

INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (216, NULL, 'gn_synthese.get_default_nomenclature_value(''STATUT_BIO'
↳'::character varying)', 'id_nomenclature_bio_status', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (218, NULL, 'gn_synthese.get_default_nomenclature_value(''NATURALITE'
↳'::character varying)', 'id_nomenclature_naturalness', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (220, NULL, 'gn_synthese.get_default_nomenclature_value(''STATUT_VALID'
↳'::character varying)', 'id_nomenclature_valid_status', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (221, NULL, 'gn_synthese.get_default_nomenclature_value(''NIV_PRECIS'
↳'::character varying)', 'id_nomenclature_diffusion_level', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (223, NULL, 'gn_synthese.get_default_nomenclature_value(''SEXE'::character
↳varying)', 'id_nomenclature_sex', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (222, NULL, 'gn_synthese.get_default_nomenclature_value(''STADE_VIE'
↳'::character varying)', 'id_nomenclature_life_stage', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (224, NULL, 'gn_synthese.get_default_nomenclature_value(''OBJ_DENBR'
↳'::character varying)', 'id_nomenclature_obj_count', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (226, NULL, 'gn_synthese.get_default_nomenclature_value(''SENSIBILITE'
↳'::character varying)', 'id_nomenclature_sensitivity', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (225, NULL, 'gn_synthese.get_default_nomenclature_value(''TYP_DENBR'
↳'::character varying)', 'id_nomenclature_type_count', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (227, NULL, 'gn_synthese.get_default_nomenclature_value(''STATUT_OBS'
↳'::character varying)', 'id_nomenclature_observation_status', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (228, NULL, 'gn_synthese.get_default_nomenclature_value(''DEE_FLOU'
↳'::character varying)', 'id_nomenclature_blurring', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (230, NULL, 'gn_synthese.get_default_nomenclature_value(''TYP_INF_GEO'
↳'::character varying)', 'id_nomenclature_info_geo_type', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (229, NULL, 'gn_synthese.get_default_nomenclature_value(''STATUT_SOURCE'
↳'::character varying)', 'id_nomenclature_source_status', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (233, 'cd_nom', NULL, 'cd_nom', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (237, NULL, 'NULL', 'digital_proof', 'text', NULL, 1);

```

(continues on next page)



(continued from previous page)

```

INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (238, NULL, 'NULL', 'non_digital_proof', 'text', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (239, 'altitude_retenue', NULL, 'altitude_min', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (240, 'altitude_retenue', NULL, 'altitude_max', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (244, 'dateobs', NULL, 'date_min', 'timestamp without time zone', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (245, 'dateobs', NULL, 'date_max', 'timestamp without time zone', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (246, NULL, 'NULL', 'validator', 'character varying', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (248, NULL, 'NULL', 'observers', 'character varying', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (247, NULL, 'NULL', 'validation_comment', 'text', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (250, NULL, 'gn_synthese.get_default_nomenclature_value('METH_DETERMIN
→':character varying)', 'id_nomenclature_determination_method', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (252, NULL, 'now()', 'meta_validation_date', 'timestamp without time zone',
→NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (253, NULL, 'now()', 'meta_create_date', 'timestamp without time zone', NULL,
→1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (254, NULL, 'now()', 'meta_update_date', 'timestamp without time zone', NULL,
→1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (255, NULL, 'c', 'last_action', 'character', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (235, NULL, 'gn_commons.get_default_parameter('taxref_version',
→NULL)::character varying', 'meta_v_taxref', 'character varying', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (251, 'remarques', NULL, 'comments', 'text', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (231, 'effectif_total', NULL, 'count_min', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (232, 'effectif_total', NULL, 'count_max', 'integer', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
→default_value, target_field, target_field_type, field_comments, id_matching_table)
→VALUES (258, 'taxon_saisi', NULL, 'nom_cite', 'character varying', NULL, 1);

```

(continues on next page)

(continued from previous page)

```

INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (249, NULL, 'u.nom_role || ' ' || u.prenom_role', 'determiner', 'character_
↳varying', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (234, 'taxon_saisi', NULL, 'nom_cite', 'character varying', NULL, 1);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (256, 'id_data', NULL, 'entity_source_pk_value', 'integer', NULL, 2);
INSERT INTO gn_imports.matching_fields (id_matching_field, source_field, source_
↳default_value, target_field, target_field_type, field_comments, id_matching_table)
↳VALUES (257, 'observateurs', NULL, 'id_role', 'integer', NULL, 2);
INSERT INTO gn_imports.matching_geoms (id_matching_geom, source_x_field, source_y_
↳field, source_geom_field, source_geom_format, source_srid, target_geom_field,
↳target_geom_srid, geom_comments, id_matching_table) VALUES (1, 'x', 'y', NULL, 'xy',
↳2154, 'the_geom_local', 2154, NULL, 1);
INSERT INTO gn_imports.matching_geoms (id_matching_geom, source_x_field, source_y_
↳field, source_geom_field, source_geom_format, source_srid, target_geom_field,
↳target_geom_srid, geom_comments, id_matching_table) VALUES (2, NULL, NULL, 'POINT(6.
↳064544 44.28787)', 'wkt', 4326, 'the_geom_4326', 4326, NULL, 1);
INSERT INTO gn_imports.matching_geoms (id_matching_geom, source_x_field, source_y_
↳field, source_geom_field, source_geom_format, source_srid, target_geom_field,
↳target_geom_srid, geom_comments, id_matching_table) VALUES (1, 'x', 'y', NULL, 'xy',
↳4326, 'the_geom_point', 4326, NULL, 1);

SELECT pg_catalog.setval('gn_imports.matching_fields_id_matching_field_seq', 258,
↳true);
SELECT pg_catalog.setval('gn_imports.matching_geoms_id_matching_geom_seq', 3, true);
SELECT pg_catalog.setval('gn_imports.matching_tables_id_matching_table_seq', 2, true);

-----
--IMPORT DATA--
-----

--autogenerated query by
--SELECT gn_imports.fct_generate_import_query('gn_imports.testimport', 'gn_synthese.
↳cor_observer_synthese');
INSERT INTO gn_synthese.synthese(
unique_id_sinp
,unique_id_sinp_grp
,id_nomenclature_exist_proof
,entity_source_pk_value
,id_dataset
,id_source
,id_nomenclature_grp_typ
,id_nomenclature_geo_object_nature
,id_nomenclature_obs_meth
,id_nomenclature_obs_technique
,id_nomenclature_bio_condition
,id_nomenclature_bio_status
,id_nomenclature_naturalness
,id_nomenclature_valid_status
,id_nomenclature_diffusion_level
,id_nomenclature_sex
,id_nomenclature_life_stage
,id_nomenclature_obj_count
,id_nomenclature_sensitivity

```

(continues on next page)

(continued from previous page)

```

, id_nomenclature_type_count
, id_nomenclature_observation_status
, id_nomenclature_blurring
, id_nomenclature_info_geo_type
, id_nomenclature_source_status
, cd_nom
, digital_proof
, non_digital_proof
, altitude_min
, altitude_max
, date_min
, date_max
, validator
, observers
, validation_comment
, id_nomenclature_determination_method
, meta_validation_date
, meta_create_date
, meta_update_date
, last_action
, meta_v_taxref
, comments
, count_min
, count_max
, nom_cite
)
SELECT
uuid_generate_v4()::uuid AS unique_id_sinp
, uuid_generate_v4()::uuid AS unique_id_sinp_grp
, gn_synthese.get_default_nomenclature_value('PREUVE_EXIST'::character_
↪varying)::integer AS id_nomenclature_exist_proof
, a.id_data::character_varying AS entity_source_pk_value
, a.id_lot::integer AS id_dataset
, a.id_source::integer AS id_source
, gn_synthese.get_default_nomenclature_value('TYP_GRP'::character_varying)::integer AS
↪id_nomenclature_grp_typ
, gn_synthese.get_default_nomenclature_value('NAT_OBJ_GEO'::character_
↪varying)::integer AS id_nomenclature_geo_object_nature
, gn_synthese.get_default_nomenclature_value('METH_OBS'::character_varying)::integer_
↪AS id_nomenclature_obs_meth
, gn_synthese.get_default_nomenclature_value('TECHNIQUE_OBS'::character_
↪varying)::integer AS id_nomenclature_obs_technique
, gn_synthese.get_default_nomenclature_value('ETA_BIO'::character_varying)::integer AS
↪id_nomenclature_bio_condition
, gn_synthese.get_default_nomenclature_value('STATUT_BIO'::character_varying)::integer_
↪AS id_nomenclature_bio_status
, gn_synthese.get_default_nomenclature_value('NATURALITE'::character_varying)::integer_
↪AS id_nomenclature_naturalness
, gn_synthese.get_default_nomenclature_value('STATUT_VALID'::character_
↪varying)::integer AS id_nomenclature_valid_status
, gn_synthese.get_default_nomenclature_value('NIV_PRECIS'::character_varying)::integer_
↪AS id_nomenclature_diffusion_level
, gn_synthese.get_default_nomenclature_value('SEXE'::character_varying)::integer AS id_
↪nomenclature_sex
, gn_synthese.get_default_nomenclature_value('STADE_VIE'::character_varying)::integer_
↪AS id_nomenclature_life_stage
, gn_synthese.get_default_nomenclature_value('OBJ_DENBR'::character_varying)::integer_
↪AS id_nomenclature_obj_count

```

(continues on next page)

(continued from previous page)

```

,gn_synthese.get_default_nomenclature_value('SENSIBILITE'::character_
↪varying)::integer AS id_nomenclature_sensitivity
,gn_synthese.get_default_nomenclature_value('TYP_DENBR'::character varying)::integer_
↪AS id_nomenclature_type_count
,gn_synthese.get_default_nomenclature_value('STATUT_OBS'::character varying)::integer_
↪AS id_nomenclature_observation_status
,gn_synthese.get_default_nomenclature_value('DEE_FLOU'::character varying)::integer_
↪AS id_nomenclature_blurring
,gn_synthese.get_default_nomenclature_value('TYP_INF_GEO'::character_
↪varying)::integer AS id_nomenclature_info_geo_type
,gn_synthese.get_default_nomenclature_value('STATUT_SOURCE'::character_
↪varying)::integer AS id_nomenclature_source_status
,a.cd_nom::integer AS cd_nom
,NULL::text AS digital_proof
,NULL::text AS non_digital_proof
,a.altitude_retenue::integer AS altitude_min
,a.altitude_retenue::integer AS altitude_max
,a.dateobs::timestamp without time zone AS date_min
,a.dateobs::timestamp without time zone AS date_max
,NULL::character varying AS validator
,NULL::character varying AS observers
,NULL::text AS validation_comment
,gn_synthese.get_default_nomenclature_value('METH_DETERMIN'::character_
↪varying)::integer AS id_nomenclature_determination_method
,now()::timestamp without time zone AS meta_validation_date
,now()::timestamp without time zone AS meta_create_date
,now()::timestamp without time zone AS meta_update_date
,'c'::character AS last_action
,gn_commons.get_default_parameter('taxref_version',NULL)::character_
↪varying::character varying AS meta_v_taxref
,a.remarques::text AS comments
,a.effectif_total::integer AS count_min
,a.effectif_total::integer AS count_max
,taxon_saisi
FROM gn_imports.testimport a
;

--autogenerated query by
--SELECT gn_imports.fct_generate_import_query('gn_imports.testimport', 'gn_synthese.
↪cor_observer_synthese');
INSERT INTO gn_synthese.cor_observer_synthese(
id_role
,id_synthese
)
  SELECT
a.observateurs::integer AS id_role
,s.id_synthese::integer AS id_synthese
FROM gn_imports.testimport a
--self addition
JOIN gn_synthese.synthese s ON s.entity_source_pk_value::integer = a.id_data
WHERE s.id_source = 4;
;

```



## DEVELOPPEMENT

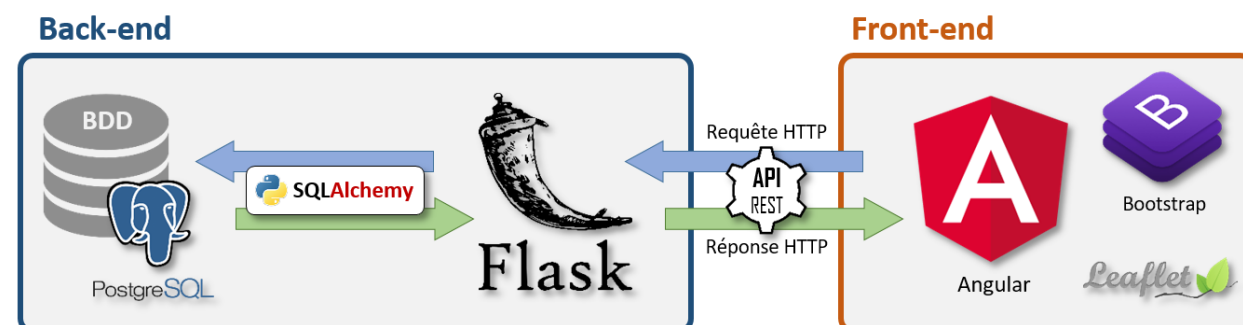
## 6.1 Général

GeoNature a été développé par Gil Deluermoz depuis 2010 avec PHP/Symfony/ExtJS.

En 2017, les parcs nationaux français ont décidé de refondre GeoNature complètement avec une nouvelle version (V2) réalisée en Python/Flask/Angular.

Mainteneurs :

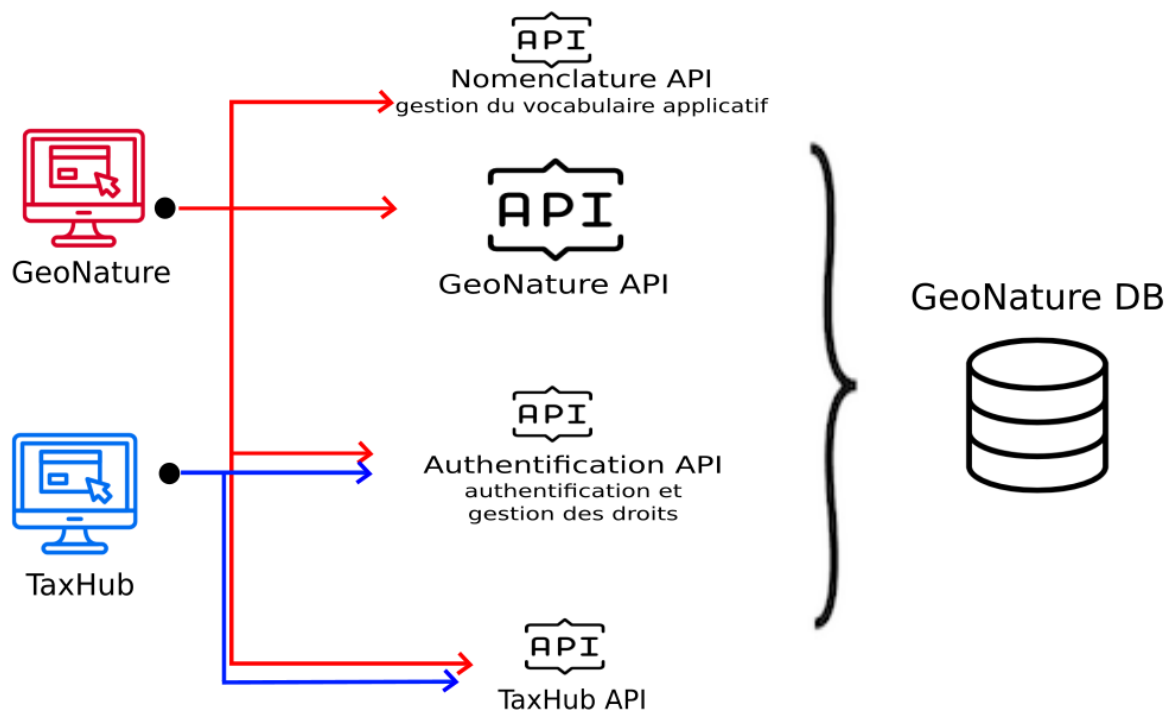
- Elie BOUTTIER (PnEcrins)
- Theo LECHEMIA (PnEcrins)
- Amandine SAHL (PnCevennes)
- Camille MONCHICOURT (PnEcrins)



## 6.2 API

GeoNature utilise :

- l'API de TaxHub (recherche taxon, règne et groupe d'un taxon...)
- l'API du sous-module Nomenclatures (typologies et listes déroulantes)
- l'API du sous-module d'authentification de UsersHub (login/logout, récupération du CRUVED d'un utilisateur)
- l'API de GeoNature (get, post, update des données des différents modules, métadonnées, intersections géographiques, exports...)



### 6.2.1 Liste des routes

Resource	Operation	Description
	<i>POST /admin/bibnomenclaturestypesadmin/ajax/update/</i>	
	<i>POST /admin/tnomenclaturesadmin/ajax/update/</i>	
	<i>POST /admin/tadditionalfields/ajax/update/</i>	
	<i>GET /admin/bibnomenclaturestypesadmin/details/</i>	
	<i>POST /admin/bibnomenclaturestypesadmin/action/</i>	
	<i>POST /admin/bibnomenclaturestypesadmin/delete/</i>	
	<i>GET /admin/bibnomenclaturestypesadmin/edit/</i>	
	<i>POST /admin/bibnomenclaturestypesadmin/edit/</i>	
	<i>GET /admin/bibnomenclaturestypesadmin/new/</i>	
	<i>POST /admin/bibnomenclaturestypesadmin/new/</i>	
	<i>GET /admin/tnomenclaturesadmin/details/</i>	
	<i>POST /admin/tnomenclaturesadmin/action/</i>	
	<i>POST /admin/tnomenclaturesadmin/delete/</i>	
	<i>GET /admin/tnomenclaturesadmin/edit/</i>	
	<i>POST /admin/tnomenclaturesadmin/edit/</i>	
	<i>GET /admin/tnomenclaturesadmin/new/</i>	
	<i>POST /admin/tnomenclaturesadmin/new/</i>	
	<i>GET /admin/tadditionalfields/details/</i>	
	<i>POST /admin/tadditionalfields/action/</i>	

continues on next page



Table 1 – continued from previous page

Resource	Operation	Description
	<i>POST /admin/tadditionalfields/delete/</i>	
	<i>GET /admin/tadditionalfields/edit/</i>	
	<i>POST /admin/tadditionalfields/edit/</i>	
	<i>GET /admin/tadditionalfields/new/</i>	
	<i>POST /admin/tadditionalfields/new/</i>	
	<i>PUT /users/password/change</i>	
	<i>PUT /users/password/new</i>	
	<i>POST /users/login/recovery</i>	
	<i>GET /pypn/register/test_uh</i>	
	<i>GET /monitorings/test_permission</i>	
	<i>GET /monitorings/routes_list</i>	
	<i>GET /monitorings/modules</i>	
	<i>GET /synthese/taxa_distribution</i>	
	<i>POST /synthese/reports</i>	
	<i>GET /occtax/vreleveocctax</i>	
	<i>GET /admin/bibnomenclaturestypesadmin/</i>	
	<i>GET /admin/tnomenclaturesadmin/</i>	
	<i>GET /admin/tadditionalfields/</i>	
	<i>GET /users/confirmation</i>	
	<i>POST /users/inscription</i>	
	<i>PUT /users/role</i>	
	<i>GET /meta/acquisition_frameworks</i>	
	<i>POST /meta/upload_canvas</i>	
	<i>GET /monitorings/exports/csv/(module_code)/(type)/(method)/(jd)</i>	
	<i>POST /monitorings/exports/pdf/(module_code)/(object_type)/(int:id)</i>	
	<i>GET /monitorings/object/module</i>	
	<i>GET /monitorings/object/(string:module_code)/(string:object_type)</i>	
	<i>GET /monitorings/object/(string:module_code)/(string:object_type)/(int:id)</i>	
	<i>GET /monitorings/util/nomenclature/(string:code_nomenclature_type)/(string:cd_nomenclature)</i>	
	<i>GET /monitorings/util/init_data/(string:module_code)</i>	
	<i>POST /occtax/releve/(int:id_releve)/occurrence</i>	
	<i>POST /occtax/only/releve/(int:id_releve)</i>	
	<i>GET /meta/acquisition_frameworks/export_pdf/(id_acquisition_framework)</i>	
	<i>POST /pypn/register/post_usershub/(string:type_action)</i>	
	<i>GET /meta/dataset/export_pdf/(id_dataset)</i>	
	<i>GET /monitorings/config</i>	
	<i>GET /monitorings/config/(string:module_code)</i>	
	<i>GET /monitorings/module/(value)</i>	
	<i>GET /monitorings/util/(string:type_util)/(int:id)</i>	
	<i>GET /monitorings/util/(string:type_util)/(string:ids)</i>	
	<i>GET /validation/date/(uuid:uuid)</i>	
	<i>GET /synthese/observation_count_per_column/(column)</i>	
	<i>GET /synthese/vsynthese/(id_synthese)</i>	
	<i>PUT /synthese/reports/(int:id_report)</i>	
	<i>POST /occtax/occurrence/(int:id_occurrence)</i>	
	<i>GET /admin/static/(path:filename)</i>	
Commons	<i>GET /gn_commons/list/parameters</i>	
	<i>GET /gn_commons/t_mobile_apps</i>	

continues on next page

Table 1 – continued from previous page

Resource	Operation	Description
	<i>GET /gn_commons/modules</i>	
	<i>PUT /gn_commons/media/(int:id_media)</i>	
	<i>PUT /gn_commons/media</i>	
	<i>POST /gn_commons/media/(int:id_media)</i>	
	<i>POST /gn_commons/media</i>	
	<i>GET /gn_commons/media/thumbnails/(int:id_media)/(int:size)</i>	
	<i>GET /gn_commons/médias/(string:uuid_attached_row)</i>	
	<i>GET /gn_commons/media/(int:id_media)</i>	
	<i>DELETE /gn_commons/media/(int:id_media)</i>	
Generic	<i>GET /config</i>	
Habref	<i>GET /habref/habitats/autocomplete</i>	
	<i>GET /habref/typo</i>	
	<i>GET /habref/correspondance/(int:cd_hab)</i>	
	<i>GET /habref/habitat/(int:cd_hab)</i>	
	<i>GET /habref/search/(field)/(ilike)</i>	
Metadata	<i>GET /meta/list/acquisition_frameworks</i>	
	<i>POST /meta/acquisition_framework</i>	
	<i>GET /meta/sensi_report</i>	
	<i>GET /meta/uuid_report</i>	
	<i>GET /meta/datasets</i>	
	<i>POST /meta/dataset</i>	
	<i>GET /meta/acquisition_framework/publish/(int:af_id)</i>	
	<i>GET /meta/acquisition_framework/(id_acquisition_framework)/stats</i>	
	<i>GET /meta/acquisition_framework/(id_acquisition_framework)/bbox</i>	
	<i>DELETE /meta/acquisition_framework/(int:af_id)</i>	
	<i>POST /meta/acquisition_framework/(int:id_acquisition_framework)</i>	
	<i>GET /meta/acquisition_framework/(id_acquisition_framework)</i>	
	<i>GET /meta/dataset/(int:id_dataset)</i>	
	<i>DELETE /meta/dataset/(int:ds_id)</i>	
	<i>PATCH /meta/dataset/(int:id_dataset)</i>	
	<i>POST /meta/dataset/(int:id_dataset)</i>	
Monitoring	<i>GET /gn_monitoring/siteslist</i>	
	<i>GET /gn_monitoring/siteslist/(int:id_site)</i>	
	<i>GET /gn_monitoring/siteareas/(int:id_site)</i>	
Nomenclatures	<i>GET /nomenclatures/nomenclatures</i>	
	<i>GET /nomenclatures/nomenclature/(int:id_type)</i>	
	<i>GET /nomenclatures/nomenclature/(string:code_type)</i>	
Occhab	<i>POST /occhab/station</i>	
Occhab	<i>GET /occhab/defaultNomenclatures</i>	
	<i>GET /occhab/stations</i>	
	<i>POST /occhab/export_stations/(export_format)</i>	
	<i>GET /occhab/station/(int:id_station)</i>	
	<i>DELETE /occhab/station/(int:id_station)</i>	
Occtax	<i>POST /occtax/only/releve</i>	Post one Occtax data (Releve + Occurrence + Counting)
	<i>GET /occtax/defaultNomenclatures</i>	
	<i>GET /occtax/occurrences</i>	
	<i>GET /occtax/relevés</i>	

continues on next page

Table 1 – continued from previous page

Resource	Operation	Description
	<i>POST /occtax/releve</i>	Post one Occtax data (Releve + Occurrence + Counting)
	<i>GET /occtax/export</i>	Export data from pr_occtax.v_export_occtax
	<i>DELETE /occtax/releve/occurrence_counting/(int:id_count)</i>	
	<i>DELETE /occtax/occurrence/(int:id_occ)</i>	
	<i>GET /occtax/counting/(int:id_counting)</i>	
	<i>GET /occtax/releve/(int:id_releve)</i>	
	<i>DELETE /occtax/releve/(int:id_releve)</i>	
Permissions	<i>GET /permissions/logout_cruved</i>	
	<i>GET /permissions/cruved</i>	
Profiles	<i>POST /gn_profiles/check_observation</i>	
	<i>GET /gn_profiles/cor_taxon_phenology/(int:cd_ref)</i>	
	<i>GET /gn_profiles/consistancy_data/(id_synthese)</i>	
	<i>GET /gn_profiles/valid_profile/(int:cd_ref)</i>	
Ref Geo	<i>GET /geo/municipalities</i>	
	<i>POST /geo/area_size</i>	
	<i>POST /geo/altitude</i>	
	<i>POST /geo/areas</i>	
	<i>GET /geo/areas</i>	
	<i>POST /geo/info</i>	
Synthese	<i>GET /synthese/defaultsNomenclatures</i>	
	<i>POST /synthese/export_observations</i>	
	<i>GET /synthese/taxons_autocomplete</i>	
	<i>GET /synthese/observation_count</i>	
	<i>GET /synthese/observations_bbox</i>	
	<i>GET /synthese/export_metadata</i>	
	<i>POST /synthese/export_metadata</i>	
	<i>POST /synthese/export_statuts</i>	
	<i>POST /synthese/export_taxons</i>	
	<i>GET /synthese/general_stats</i>	
	<i>GET /synthese/taxons_tree</i>	
	<i>GET /synthese/color_taxon</i>	
	<i>GET /synthese/taxa_count</i>	
	<i>GET /synthese/for_web</i>	Get filtered observations
	<i>POST /synthese/for_web</i>	
	<i>GET /synthese/sources</i>	
	<i>GET /synthese</i>	Deprecated
User	<i>GET /gn_auth/logout_cruved</i>	
	<i>GET /gn_auth/login_cas</i>	
	<i>POST /gn_auth/login_cas</i>	
	<i>GET /users/organisms_dataset_actor</i>	
	<i>GET /users/organisms</i>	
	<i>GET /users/roles</i>	
	<i>GET /users/menu_from_code/(string:code_liste)</i>	
	<i>GET /users/menu/(int:id_menu)</i>	
	<i>GET /users/role/(int:id_role)</i>	
Validation	<i>GET /validation</i>	
	<i>POST /validation</i>	

continues on next page

Table 1 – continued from previous page

Resource	Operation	Description
View_Permission	<i>GET /permissions_backoffice/users</i>	
	<i>GET /permissions_backoffice/other_permissions_form/id_permission/(int:id_permission)/user/(int:id_role)/filter_type/(int:id_filter_type)</i>	
	<i>GET /permissions_backoffice/other_permissions_form/user/(int:id_role)/filter_type/(int:id_filter_type)</i>	
	<i>POST /permissions_backoffice/other_permissions_form/id_permission/(int:id_permission)/user/(int:id_role)/filter_type/(int:id_filter_type)</i>	
	<i>POST /permissions_backoffice/other_permissions_form/user/(int:id_role)/filter_type/(int:id_filter_type)</i>	
	<i>GET /permissions_backoffice/cruved_form/module/(int:id_module)/role/(int:id_role)/object/(int:id_object)</i>	
	<i>GET /permissions_backoffice/cruved_form/module/(int:id_module)/role/(int:id_role)</i>	
	<i>POST /permissions_backoffice/cruved_form/module/(int:id_module)/role/(int:id_role)/object/(int:id_object)</i>	
	<i>POST /permissions_backoffice/cruved_form/module/(int:id_module)/role/(int:id_role)</i>	
	<i>GET /permissions_backoffice/filter_form/id_filter_type/(int:id_filter_type)/id_filter/(int:id_filter)</i>	
	<i>GET /permissions_backoffice/filter_form/id_filter_type/(int:id_filter_type)</i>	
	<i>POST /permissions_backoffice/filter_form/id_filter_type/(int:id_filter_type)/id_filter/(int:id_filter)</i>	
	<i>POST /permissions_backoffice/filter_form/id_filter_type/(int:id_filter_type)</i>	
	<i>GET /permissions_backoffice/filter_list/id_filter_type/(int:id_filter_type)</i>	
	<i>GET /permissions_backoffice/user_other_permissions/(id_role)</i>	
	<i>GET /permissions_backoffice/user_cruved/(id_role)</i>	
	<i>POST /permissions_backoffice/filter/(id_filter)</i>	

## 6.2.2 Documentation des routes

**POST /admin/bibnomenclaturestypesadmin/ajax/update/**

Edits a single column of a record in list view.

**POST /admin/tnomenclaturesadmin/ajax/update/**

Edits a single column of a record in list view.

**POST /admin/tadditionalfields/ajax/update/**

Edits a single column of a record in list view.

**GET /gn\_commons/list/parameters**

Get all parameters from gn\_commons.t\_parameters

**GET /habref/habitats/autocomplete**

Get all habref items of a list for autocomplete

### Query Parameters

- **int** (*limit*) – the id of the habref list
- **str** (*search\_name*) – the pattern to filter with
- **int** – filter by typology
- **int** – number of results, default = 20

**Returns** Array<AutoCompleteHabitat>

**POST /occtax/only/releve**

Post one Occtax data (Releve + Occurrence + Counting)

**Request JSON object:**

```
{
  "geometry":
    { "type": "Point",
      "coordinates": [0.9008789062500001, 47.14489748555398] },
}
```

(continues on next page)

(continued from previous page)

```

    "properties":
    {
        "id_releve_occtax":null,"id_dataset":1,"id_digitiser":1,"date_min":"2019-
↪05-09","date_max":"2019-05-09","hour_min":null,"hour_max":null,"altitude_min
↪":null,"altitude_max":null,"meta_device_entry":"web","comment":null,"id_
↪nomenclature_obs_technique":316,"observers":[1],"observers_txt":null,"id_
↪nomenclature_grp_typ":132,
        "t_occurrences_occtax":[{
            "id_releve_occtax":null,"id_occurrence_occtax":null,"id_nomenclature_
↪obs_technique":41,"id_nomenclature_bio_condition":157,"id_nomenclature_bio_
↪status":29,"id_nomenclature_naturalness":160,"id_nomenclature_exist_proof":81,
↪"id_nomenclature_observation_status":88,"id_nomenclature_blurring":175,"id_
↪nomenclature_source_status":75,"determiner":null,"id_nomenclature_determination_
↪method":445,"cd_nom":67111,"nom_cite":"Ablette = <i> Alburnus alburnus_
↪(Linnaeus, 1758)</i> - [ES - 67111]","meta_v_taxref":null,"sample_number_proof
↪":null,"comment":null,
            "cor_counting_occtax":[{
                "id_counting_occtax":null,"id_nomenclature_life_stage":1,"id_
↪nomenclature_sex":171,"id_nomenclature_obj_count":146,"id_nomenclature_type_
↪count":94,"id_occurrence_occtax":null,"count_min":1,"count_max":1
            }]
        }]
    }
}

```

**Returns** GeoJson<TRelevésOccurrence>

**GET** /admin/bibnomenclaturestypesadmin/details/  
Details model view

**POST** /admin/bibnomenclaturestypesadmin/action/  
Mass-model action view.

**POST** /admin/bibnomenclaturestypesadmin/delete/  
Delete model view. Only POST method is allowed.

**GET** /admin/bibnomenclaturestypesadmin/edit/  
Edit model view

**POST** /admin/bibnomenclaturestypesadmin/edit/  
Edit model view

**GET** /admin/bibnomenclaturestypesadmin/new/  
Create model view

**POST** /admin/bibnomenclaturestypesadmin/new/  
Create model view

**GET** /admin/tnomenclaturesadmin/details/  
Details model view

**POST** /admin/tnomenclaturesadmin/action/  
Mass-model action view.

**POST** /admin/tnomenclaturesadmin/delete/  
Delete model view. Only POST method is allowed.

**GET** /admin/tnomenclaturesadmin/edit/  
Edit model view

**POST** /admin/tnomenclaturesadmin/edit/  
Edit model view

**GET** /admin/tnomenclaturesadmin/new/  
Create model view

**POST** /admin/tnomenclaturesadmin/new/  
Create model view

**GET** /admin/tadditionalfields/details/  
Details model view

**POST** /admin/tadditionalfields/action/  
Mass-model action view.

**POST** /admin/tadditionalfields/delete/  
Delete model view. Only POST method is allowed.

**GET** /admin/tadditionalfields/edit/  
Edit model view

**POST** /admin/tadditionalfields/edit/  
Edit model view

**GET** /admin/tadditionalfields/new/  
Create model view

**POST** /admin/tadditionalfields/new/  
Create model view

**PUT** /users/password/change  
Modifie le mot de passe de l'utilisateur connecté et de son ancien mdp Fait appel à l'API UsersHub

**PUT** /users/password/new  
Modifie le mdp d'un utilisateur apres que celui-ci ai demander un renouvellement Necessite un token envoyer par mail a l'utilisateur

**POST** /users/login/recovery  
Call UsersHub API to create a TOKEN for a user A post\_action send an email with the user login and a link to reset its password Work only if 'ENABLE\_SIGN\_UP' is set to True

**GET** /pypn/register/test\_uh  
route pour tester le décorateur connect\_admin ainsi que les paramètres de connexion à USERSHUB:

- config['ADMIN\_APPLICATION\_LOGIN']
- config['ADMIN\_APPLICATION\_PASSWORD']

**GET** /meta/list/acquisition\_frameworks  
Get all AF with their datasets Use in metadata module for list of AF and DS Add the CRUVED permission for each row (Dataset and AD)

#### Parameters

- **info\_role** (*TRole*) – add with kwargs

#### Query Parameters

- **excluded\_fields** (*list*) – fields excluded from serialization
- **nested** (*boolean*) – Default False - serialized relationships. If false: remove add all relationships in excluded\_fields

**GET /permissions\_backoffice/users**

Render a list with all users with their number of cruved Link to edit cruved and other permissions Only display user which have profil in GeoNature and active user

**GET /nomenclatures/nomenclatures**

Route : liste des termes d'un ensemble de nomenclatures Possibilité de filtrer par regne et group2Inpn

**GET /gn\_monitoring/siteslist**

Return the sites list for an application in a dict {id\_base\_site, nom site} .. :quickref: Monitoring;

**Parameters**

- **id\_base\_site** – id of base site
- **module\_code** – code of the module
- **id\_module** – id of the module
- **base\_site\_name** – part of the name of the site
- **type** – int

**POST /gn\_profiles/check\_observation**

Check an observation with the related profile Return alert when the observation do not match the profile

**GET /monitorings/test\_permission**

tests sur les permissions

**GET /permissions/logout\_cruved**

Route to logout with cruved

To avoid multiples server call, we store the cruved in the session when the user logout we need clear the session to get the new cruved session

**GET /monitorings/routes\_list**

retourne la liste des routes du module

**GET /monitorings/modules**

Renvoie la liste des modules de suivi

**GET /permissions/cruved**

Get the cruved for a user

Params: :param user: the user who ask the route, auto kwargs via @check\_cruved\_scope :type user: User  
:param module\_code: the code of the requested module - as querystring :type module\_code: str

**Returns** dict of the CRUVED

**GET /gn\_commons/t\_mobile\_apps**

Get all mobile applications

**Query Parameters**

- **app\_code** (*str*) – the app code

**Returns** Array<dict<TMobileApps>>

**GET /gn\_commons/modules**

Return the allowed modules of user from its cruved .. :quickref: Commons;

**PUT /gn\_commons/media/ (int: id\_media)****PUT /gn\_commons/media**

Insertion ou mise à jour d'un média avec prise en compte des fichiers joints

**POST /gn\_commons/media/ (int: id\_media)**

**POST /gn\_commons/media**

Insertion ou mise à jour d'un média avec prise en compte des fichiers joints

**GET /synthese/defaultsNomenclatures**

Get default nomenclatures

**Query Parameters**

- **group2\_inpn** (*str*) –
- **regne** (*str*) –
- **organism** (*int*) –

**POST /synthese/export\_observations**

Optimized route for observations web export.

This view is customisable by the administrator Some columns are mandatory: id\_synthese, geojson and geojson\_local to generate the exported files

POST parameters: Use a list of id\_synthese (in POST parameters) to filter the v\_synthese\_for\_export\_view

**Query Parameters**

- **export\_format** (*str*) – str<'csv', 'geojson', 'shapefiles', 'gpkg'>

**GET /synthese/taxons\_autocomplete**

Autocomplete taxon for web search (based on all taxon in Synthese).

The request use trigram algorithm to get relevant results

**Query Parameters**

- **search\_name** (*str*) – the search name (use sql ilike statement and puts “%” for spaces)
- **regne** (*str*) – filter with kingdom

:query str group2\_inpn : filter with INPN group 2

**GET /synthese/observation\_count**

Get observations found in a given dataset

id\_dataset: *int* (query parameter)

**count:** *int*: the number of observation

**GET /synthese/observations\_bbox**

Get bbbox of observations

id\_dataset: *int*: (query parameter)

**bbbox: geojson:** the bounding box in geojson

**GET /synthese/taxa\_distribution**

Get taxa distribution for a given dataset or acquisition framework and grouped by a certain taxa rank

**GET /synthese/export\_metadata**

Route to export the metadata in CSV

The table `synthese` is join with `gn_synthese.v_metadata_for_export` The column `jdd_id` is mandatory in the view `gn_synthese.v_metadata_for_export`

POST parameters: Use a list of id\_synthese (in POST parameters) to filter the v\_synthese\_for\_export\_view

**POST /synthese/export\_metadata**

Route to export the metadata in CSV



The table `synthese` is join with `gn_synthese.v_metadata_for_export`. The column `jdd_id` is mandatory in the view `gn_synthese.v_metadata_for_export`.

POST parameters: Use a list of `id_synthese` (in POST parameters) to filter the `v_synthese_for_export_view`.

#### POST /synthese/export\_statuts

Route to get all the protection status of a `synthese` search.

Get the CRUVED from 'R' action because we don't give observations X/Y but only `statuts` and to be constant with the data displayed in the web interface.

##### Parameters:

- HTTP-GET: the same that the `/synthese` endpoint (all the filter in web app)

#### POST /synthese/export\_taxons

Optimized route for taxon web export.

This view is customisable by the administrator. Some columns are mandatory: `cd_ref`.

POST parameters: Use a list of `cd_ref` (in POST parameters) to filter the `v_synthese_taxon_for_export_view`.

##### Query Parameters

- `export_format` (*str*) – `str<'csv'>`

#### GET /synthese/general\_stats

Return stats about `synthese`.

#### GET /synthese/taxons\_tree

Get taxon tree.

#### GET /synthese/color\_taxon

Get color of taxon in areas (vue `synthese.v_color_taxon_area`).

##### Query Parameters

- `code_area_type` (*str*) – Type area code (`ref_geo.bib_areas_types.type_code`)
- `id_area` (*int*) – Id of area (`ref_geo.l_areas.id_area`)
- `cd_nom` (*int*) – taxon code (`taxonomie.taxref.cd_nom`)

Those three parameters can be multiples :returns: `Array<dict<VColorAreaTaxon>>`

#### GET /synthese/taxa\_count

Get taxa count in `synthese` filtering with generic parameters.

`id_dataset`: *int* (query parameter)

`count`: *int*: the number of taxon

#### GET /synthese/for\_web

Optimized route to serve data for the frontend with all filters.

Query filtered by any filter, returning all the fields of the view `v_synthese_for_export`:

```
properties = {
    "id": r["id_synthese"],
    "date_min": str(r["date_min"]),
    "cd_nom": r["cd_nom"],
    "nom_vern_or_lb_nom": r["nom_vern"] if r["nom_vern"] else r["lb_nom"],
    "lb_nom": r["lb_nom"],
```

(continues on next page)

(continued from previous page)

```

    "dataset_name": r["dataset_name"],
    "observers": r["observers"],
    "url_source": r["url_source"],
    "unique_id_sinp": r["unique_id_sinp"],
    "entity_source_pk_value": r["entity_source_pk_value"],
}
geojson = json.loads(r["st_asgeojson"])
geojson["properties"] = properties

```

### Parameters

- **info\_role** (*str*) – Role used to get the associated filters, **TBC**

### Query Parameters

- **limit** (*str*) – Limit number of synthese returned. Defaults to NB\_MAX\_OBS\_MAP.
- **cd\_ref** (*str*) – Filter by TAXREF cd\_ref attribute
- **taxonomy\_group2\_inpn** (*str*) – Filter by TAXREF group2\_inpn attribute
- **taxonomy\_id\_hab** (*str*) – Filter by TAXREF id\_habitat attribute
- **taxonomy\_lr** (*str*) – Filter by TAXREF cd\_ref attribute
- **taxhub\_attribut\*** (*str*) – Generic TAXREF filter, given attribute & value
- **observers** (*str*) – Filter on observer
- **id\_organism** (*str*) – Filter on organism
- **date\_min** (*str*) – Start date
- **date\_max** (*str*) – End date
- **id\_acquisition\_framework** (*str*) – *tbd*
- **geoIntersection** (*str*) – Intersect with the geom send from the map
- **period\_start** (*str*) – *tbd*
- **period\_end** (*str*) – *tbd*
- **area\*** (*str*) – Generic filter on area
- **\*** (*str*) – Generic filter, given by colname & value

### Response JSON Array of Objects

- **data** (*array*) – Array of synthese with geojson key, see above
- **nb\_total** (*int*) – Number of observations
- **nb\_obs\_limited** (*bool*) – Is number of observations capped

### POST /synthese/for\_web

Optimized route to serve data for the frontend with all filters.

Query filtered by any filter, returning all the fields of the view v\_synthese\_for\_export:

```

properties = {
    "id": r["id_synthese"],
    "date_min": str(r["date_min"]),
    "cd_nom": r["cd_nom"],

```

(continues on next page)

(continued from previous page)

```

"nom_vern_or_lb_nom": r["nom_vern"] if r["nom_vern"] else r["lb_nom"],
"lb_nom": r["lb_nom"],
"dataset_name": r["dataset_name"],
"observers": r["observers"],
"url_source": r["url_source"],
"unique_id_sinp": r["unique_id_sinp"],
"entity_source_pk_value": r["entity_source_pk_value"],
}
geojson = json.loads(r["st_asgeojson"])
geojson["properties"] = properties

```

**Parameters**

- **info\_role** (*str*) – Role used to get the associated filters, **TBC**

**Query Parameters**

- **limit** (*str*) – Limit number of synthese returned. Defaults to NB\_MAX\_OBS\_MAP.
- **cd\_ref** (*str*) – Filter by TAXREF cd\_ref attribute
- **taxonomy\_group2\_inpn** (*str*) – Filter by TAXREF group2\_inpn attribute
- **taxonomy\_id\_hab** (*str*) – Filter by TAXREF id\_habitat attribute
- **taxonomy\_lr** (*str*) – Filter by TAXREF cd\_ref attribute
- **taxhub\_attribut\*** (*str*) – Generic TAXREF filter, given attribute & value
- **observers** (*str*) – Filter on observer
- **id\_organism** (*str*) – Filter on organism
- **date\_min** (*str*) – Start date
- **date\_max** (*str*) – End date
- **id\_acquisition\_framework** (*str*) – *tbd*
- **geoIntersection** (*str*) – Intersect with the geom send from the map
- **period\_start** (*str*) – *tbd*
- **period\_end** (*str*) – *tbd*
- **area\*** (*str*) – Generic filter on area
- **\*** (*str*) – Generic filter, given by colname & value

**Response JSON Array of Objects**

- **data** (*array*) – Array of synthese with geojson key, see above
- **nb\_total** (*int*) – Number of observations
- **nb\_obs\_limited** (*bool*) – Is number of observations capped

**GET /synthese/sources**

Get all sources.

**POST /synthese/reports**

Create a report (e.g report) for a given synthese id

**report: json:** Every occurrence's report

**GET /gn\_auth/logout\_cruved**

Route to logout with cruved To avoid multiples server call, we store the cruved in the session when the user logout we need clear the session to get the new cruved session

**GET /gn\_auth/login\_cas**

Login route with the INPN CAS

**POST /gn\_auth/login\_cas**

Login route with the INPN CAS

**GET /occtax/defaultNomenclatures**

Get default nomenclatures define in occtax module

**Returns** dict: {'MODULE\_CODE': 'ID\_NOMENCLATURE'}

**GET /occhab/defaultNomenclatures**

Get default nomenclatures define in occhab module

**Returns** dict: {'MODULE\_CODE': 'ID\_NOMENCLATURE'}

**GET /occtax/vreleveocctax**

Deprecated

**GET /occtax/occurrences**

Get all Occurrences

**Returns** dict<TOccurrencesOccurrence>

**GET /occhab/stations**

Get all stations with their hab

**GET /occtax/relevés**

Route for map list web interface

**POST /occhab/station**

Post one occhab station (station + habitats)

Post one occhab station (station + habitats)

**Returns** GeoJson<TStationsOcchab>

**POST /occtax/releve**

Route utilisée depuis l'appli mobile => dépréciée et non utilisée par l'appli web Post one Occtax data (Releve + Occurrence + Counting)

**Request JSON object:**

```
{
  "geometry":
    { "type": "Point",
      "coordinates": [0.9008789062500001, 47.14489748555398] },
  "properties":
    {
      "id_releve_occtax": null, "id_dataset": 1, "id_digitiser": 1, "date_min": "2019-
      ↪ 05-09", "date_max": "2019-05-09", "hour_min": null, "hour_max": null, "altitude_min
      ↪ ": null, "altitude_max": null, "meta_device_entry": "web", "comment": null, "id_
      ↪ nomenclature_obs_technique": 316, "observers": [1], "observers_txt": null, "id_
      ↪ nomenclature_grp_typ": 132,
      "t_occurrences_occtax": [ {
        "id_releve_occtax": null, "id_occurrence_occtax": null, "id_nomenclature_
        ↪ obs_technique": 41, "id_nomenclature_bio_condition": 157, "id_nomenclature_bio_
        ↪ status": 29, "id_nomenclature_naturalness": 160, "id_nomenclature_exist_proof": 81,
        ↪ "id_nomenclature_observation_status": 88, "id_nomenclature_blurring": 175, "id_
        ↪ nomenclature_source_status": 75, "determiner": null, "id_nomenclature_
        ↪ method": 445, "cd_nom": 67111, "nom_cite": "Ablette = <i> Alburnus alburnus_
        ↪ (Linnaeus, 1758)</i> — [ES — 67111]", "meta_v_taxref": null, "sample_number_proof
        ↪ ": null, "comment": null,
```

(continued from previous page)

```

    "cor_counting_occtax": [{
        "id_counting_occtax": null, "id_nomenclature_life_stage": 1, "id_
        ↪ nomenclature_sex": 171, "id_nomenclature_obj_count": 146, "id_nomenclature_type_
        ↪ count": 94, "id_occurrence_occtax": null, "count_min": 1, "count_max": 1
    }]
  }]
}
}

```

**Returns** GeoJson<TRelevancesOccurrence>

#### GET /occtax/export

Export data from pr\_occtax.v\_export\_occtax view (parameter)

##### Query Parameters

- **format** (*str*) – format of the export ('csv', 'geojson', 'shapefile', 'gpkg')

#### GET /habref/typo

Get all typology

##### Query Parameters

- **id\_list** (*int*) – return only the typology of a given id\_list

**Returns** Array<TypoRef>

#### GET /admin/bibnomenclaturestypesadmin/

List view

#### GET /users/organisms\_dataset\_actor

Get all organisms and the JDD where there are actor and where the current user has authorization with its cruved

#### GET /admin/tnomenclaturesadmin/

List view

#### GET /admin/tadditionalfields/

List view

#### GET /users/confirmation

Validate a account after a demande (this action is triggered by the link in the email) Create a personal JDD as post\_action if the parameter AUTO\_DATASET\_CREATION is set to True Fait appel à l'API UsersHub

#### POST /users/inscription

Ajoute un utilisateur à utilisateurs.temp\_user à partir de l'interface geonature Fonctionne selon l'autorisation 'ENABLE\_SIGN\_UP' dans la config. Fait appel à l'API UsersHub

#### GET /users/organisms

Get all organisms

#### GET /users/roles

Get all roles

#### PUT /users/role

Modifie le role de l'utilisateur du token en cours

#### GET /meta/acquisition\_frameworks

Get a simple list of AF without any nested relationships Use for AF select in form Get the GeoNature CRUVED

#### POST /meta/acquisition\_framework

Post one AcquisitionFramework data .. :quickref: Metadata;

**POST /meta/upload\_canvas**

Upload the canvas as a temporary image used while generating the pdf file

**GET /meta/sensi\_report**

get the UUID report of a dataset

**GET /meta/uuid\_report**

get the UUID report of a dataset

**GET /meta/datasets**

Get datasets list

**Parameters**

- **info\_role** (*TRole*) – add with kwargs

**Query Parameters**

- **active** (*boolean*) – filter on active fiel
- **id\_acquisition\_framework** (*int*) – get only dataset of given AF

**Returns** *list<TDatasets>*

**POST /meta/dataset**

Post one Dataset data .. :quickref: Metadata;

**GET /geo/municipalities**

Return the municipalities .. :quickref: Ref Geo;

**POST /geo/area\_size**

Return the area size from a given geojson

**Returns** An area size (int)

**POST /geo/altitude**

From a posted geojson get the altitude min/max

**POST /geo/areas**

From a posted geojson, the route return all the area intersected from l\_areas .. :quickref: Ref Geo;

**GET /geo/areas**

Return the areas of ref\_geo.l\_areas .. :quickref: Ref Geo;

**POST /geo/info**

From a posted geojson, the route return the municipalities intersected and the altitude min/max

**GET /validation**

Return synthese and t\_validations data filtered by form params Params must have same synthese fields names

**info\_role (User):** Information about the user asking the route. Auto add with kwargs

FeatureCollection

**POST /validation**

Return synthese and t\_validations data filtered by form params Params must have same synthese fields names

**info\_role (User):** Information about the user asking the route. Auto add with kwargs

FeatureCollection

**GET /synthese**

Return synthese row(s) filtered by form params. NOT USED ANY MORE FOR PERFORMANCE ISSUES

Deprecated since version 2?.

Use :route: /for\_web instead

Params must have same synthese fields names

### Parameters

- **info\_role** (*str*) – Role used to get the associated filters

**Returns** dict[dict, int, bool] See description above

**GET** /config

Parse and return configuration files as toml .. :quickref: Generic;

**GET** /permissions\_backoffice/other\_permissions\_form/id\_permission/ (int:   
 id\_permission) /   
 user/

int: id\_role/filter\_type/int: id\_filter\_type

**GET** /permissions\_backoffice/other\_permissions\_form/user/ (int: id\_role) /   
 filter\_type/

int: id\_filter\_type Form to define permisisions for a user expect SCOPE permissions .. :quickref:   
 View\_Permission;

**POST** /permissions\_backoffice/other\_permissions\_form/id\_permission/ (int:   
 id\_permission) /   
 user/

int: id\_role/filter\_type/int: id\_filter\_type

**POST** /permissions\_backoffice/other\_permissions\_form/user/ (int: id\_role) /   
 filter\_type/

int: id\_filter\_type Form to define permisisions for a user expect SCOPE permissions .. :quickref:   
 View\_Permission;

**GET** /permissions\_backoffice/cruved\_form/module/ (int: id\_module) /role/

int: id\_role/object/int: id\_object

**GET** /permissions\_backoffice/cruved\_form/module/ (int: id\_module) /role/

int: id\_role

**POST** /permissions\_backoffice/cruved\_form/module/ (int: id\_module) /role/

int: id\_role/object/int: id\_object

**POST** /permissions\_backoffice/cruved\_form/module/ (int: id\_module) /role/

int: id\_role

**GET** /permissions\_backoffice/filter\_form/id\_filter\_type/ (int: id\_filter\_type) /   
 id\_filter/

int: id\_filter

**GET** /permissions\_backoffice/filter\_form/id\_filter\_type/ (int: id\_filter\_type)

**POST** /permissions\_backoffice/filter\_form/id\_filter\_type/ (int: id\_filter\_type) /   
 id\_filter/

int: id\_filter

**POST** /permissions\_backoffice/filter\_form/id\_filter\_type/ (int: id\_filter\_type)

**GET** /permissions\_backoffice/filter\_list/id\_filter\_type/ (int: id\_filter\_type)

**GET** /monitorings/exports/csv/ (module\_code) /

type/method/jd Export all the observations made on a site group. Following formats are available: \* csv \*   
 geojson \* shapefile

**POST** /monitorings/exports/pdf/ (module\_code) /

object\_type/int: id Export the fiche individu as a PDF file. Need to push the map image in the post data to   
 be present in PDF. Need to set a template in sub-module.

**GET** /monitorings/object/module

**GET** /monitorings/object/ (string: module\_code) /  
string: object\_type

**GET** /monitorings/object/ (string: module\_code) /  
string: object\_type/int: id renvoie un object, à partir de type de l'object et de son id

**Parameters**

- **module\_code** – reference le module concerne
- **object\_type** – le type d'object (site, visit, observation)

:param id : l'identifiant de l'object (de id\_base\_site pour site) :type module\_code: str :type object\_type: str  
:type id: int

**Return** renvoie l'object requis

**Rtype** dict

**GET** /monitorings/util/nomenclature/ (string: code\_nomenclature\_type) /  
string: cd\_nomenclature

renvoie un champ d'un object de type nomenclature à partir de son type et de son cd\_nomenclature  
renvoie l'objet entier si field\_name renseigné en paramètre de route est 'all'

**Parameters**

- **code\_nomenclature\_type** –
- **cd\_nomenclature** –

:return object entier si field\_name = all, la valeur du champs defini par field\_name sinon

**GET** /monitorings/util/init\_data/ (string: module\_code)  
renvoie les données nomenclatures, etc à précharger par le module

**GET** /gn\_commons/media/thumbnails/ (int: id\_media) /  
int: size Retourne le thumbnail d'un media .. :quickref: Commons;

**DELETE** /occtax/releve/occurrence\_counting/ (int: id\_count)  
Delete one counting

**Params** int id\_count ID of the counting to delete

**POST** /occtax/releve/ (int: id\_releve) /occurrence  
Post one Occurrence data (Occurrence + Counting) for add to Releve

**POST** /occtax/only/releve/ (int: id\_releve)  
Post one Occurrence data (Occurrence + Counting) for add to Releve

**GET** /meta/acquisition\_frameworks/export\_pdf/ (id\_acquisition\_framework)  
Get a PDF export of one acquisition

**GET** /meta/acquisition\_framework/publish/ (int: af\_id)  
Publish an acquisition framework .. :quickref: Metadata;

**GET** /meta/acquisition\_framework/ (id\_acquisition\_framework) /stats  
Get stats from one AF .. :quickref: Metadata; :param id\_acquisition\_framework: the id\_acquisition\_framework  
:param type: int

**GET** /meta/acquisition\_framework/ (id\_acquisition\_framework) /bbox  
Get BBOX from one AF .. :quickref: Metadata; :param id\_acquisition\_framework: the  
id\_acquisition\_framework :param type: int



**POST /pypn/register/post\_usershub/ (string: *type\_action*)**  
 route generique pour appeler les routes UsersHub en tant qu'administrateur de l'appli en cours ex :  
 post\_usershub/test\_connexion appelle la route URL\_USERSHUB/api\_register/test\_connexion

**GET /meta/dataset/export\_pdf/ (id\_dataset)**  
 Get a PDF export of one dataset

**GET /permissions\_backoffice/user\_other\_permissions/ (id\_role)**  
 Get all the permissions define for a user expect SCOPE permissions

**GET /permissions\_backoffice/user\_cruved/ (id\_role)**  
 Get all scope CRUVED (with heritage) for a user in all modules

**POST /permissions\_backoffice/filter/ (id\_filter)**

**GET /nomenclatures/nomenclature/ (int: *id\_type*)**  
 => Déprécié pour des raisons de volatilité des identifiants en BD  
 Route : liste des termes d'une nomenclature basées sur les identifiants de nomenclature Possibilité de filtrer par regne et group2Inpn

**GET /nomenclatures/nomenclature/ (string: *code\_type*)**  
 Route : liste des termes d'une nomenclature basées sur le code mnemonique du type de nomenclature Possibilité de filtrer par regne et group2Inpn

**GET /gn\_monitoring/siteslist/ (int: *id\_site*)**  
 Get minimal information for a site {id\_base\_site, nom site} .. :quickref: Monitoring;

**Parameters**

- **id\_site** – id of base site
- **type** – int

**GET /gn\_monitoring/siteareas/ (int: *id\_site*)**  
 Get areas of a site from cor\_site\_area as geojson

**Parameters**

- **id\_module** (*int*) – int
- **id\_area\_type** (*int*) –

**GET /gn\_profiles/cor\_taxon\_phenology/ (int: *cd\_ref*)**  
 Get phenoliques periods for a given taxon

**GET /gn\_profiles/consistancy\_data/ (id\_synthese)**  
 Return the validation score for a synthese data

**GET /gn\_profiles/valid\_profile/ (int: *cd\_ref*)**  
 Return the profile for a cd\_ref

**GET /monitorings/config**

**GET /monitorings/config/ (string: *module\_code*)**  
 route qui renvoie la config pour un module donné

**GET /monitorings/module/ (value)**  
 Renvoie un module référencé par son champ module\_code par default cherche par id\_module on peut préciser field\_name en parametre de requete GET ?field\_name=module\_code pour avoir unmodule depuis son champs module\_code

**GET** /monitorings/util/ (string: *type\_util*) /  
**int:** *id* renvoie un champ d'un objet de type nomenclature, taxonomy, utilisateur, ... renvoie l'objet entier si field\_name renseigné en paramètre de route est 'all'

**Parameters**

- **type\_util** (*str*) – 'nomenclature' | 'taxonomy' | 'utilisateur' | etc...
- **id** (*int*) – id de l'objet requis

:return object entier si field\_name = all, la valeur du champs defini par field\_name sinon

**GET** /monitorings/util/ (string: *type\_util*) /  
**string:** *ids* variante de get\_util\_from\_id\_api pour plusieurs id renvoie un tableau de valeur (ou de dictionnaire si key est 'all')

**parametre get**

**key:** all renvoie tout l'objet sinon renvoie un champ

**separator\_out:** pour reformer une chaine de caractere a partir du tableau resultat de la requete si separator\_out == ' ', alors ['jean', 'pierre', 'paul'].join(separator\_out) -> 'jean, pierre, paul'

**Parameters**

- **type\_util** (*str*) – 'nomenclature' | 'taxonomy' | 'utilisateur'
- **ids** (*str*) – plusieurs id reliée par des '-' (ex: 1-123-3-4)

:return list si key=all ou chaine de caractere

**GET** /gn\_commons/medias/ (string: *uuid\_attached\_row*)  
Retourne des medias .. :quickref: Commons;

**GET** /gn\_commons/media/ (int: *id\_media*)  
Retourne un media .. :quickref: Commons;

**DELETE** /gn\_commons/media/ (int: *id\_media*)  
Suppression d'un media

**GET** /validation/date/ (uuid: *uuid*)  
Retourne la date de validation pour l'observation uuid

**GET** /synthese/observation\_count\_per\_column/ (*column*)  
Get observations count group by a given column

**GET** /synthese/vsynthese/ (*id\_synthese*)  
Get one synthese record for web app with all decoded nomenclature

**PUT** /synthese/reports/ (int: *id\_report*)  
Modify a report (e.g report) for a given synthese id

**report: json:** Every occurrence's report

**POST** /occhab/export\_stations/ (*export\_format*)  
Download all stations The route is in post to avoid a too large query string

**GET** /habref/correspondance/ (int: *cd\_hab*)  
Get all correspondances in other typo from a cd\_hab

**Params cd\_hab** a cd\_hab

**POST** /occtax/occurrence/ (int: *id\_occurrence*)  
Post one Occurrence data (Occurrence + Counting) for add to Releve

**DELETE** /occtax/occurrence/ (int: id\_occ)

Delete one occurrence and associated counting

**Params** int id\_occ ID of the occurrence to delete

**GET** /occtax/counting/ (int: id\_counting)

Get one counting record, with its id\_counting

**Parameters**

- **id\_counting** (int) – the pr\_occtax.cor\_counting\_occtax PK

**Returns** a dict representing a counting record

**Rtype** dict<CorCountingOccurrence>

**GET** /habref/habitat/ (int: cd\_hab)

Get one habitat with its correspondances

**Params** cd\_hab a cd\_hab

**GET** /occhab/station/ (int: id\_station)

Return one station

**Parameters**

- **id\_station** (int) – the id\_station

**Return** a dict representing one station with its habitats

:rtype dict<TStationsOcchab>

**DELETE** /occhab/station/ (int: id\_station)

Delete a station with its habitat and its observers

**GET** /habref/search/ (field) /

*ilike* Get the first 20 result of Habref table for a given field with an ilike query Use trigram algo to add relevance

**Params** field a Habref column

**Parameters**

- **ilike** – the ilike where expression to filter

:type ilike:str

**Returns** Array of dict

**GET** /occtax/releve/ (int: id\_releve)

Get one releve

**Parameters**

- **id\_releve** (int) – the id releve from pr\_occtax.t\_releve\_occtax

**Returns** Return a releve with its attached Cruved

**Rtype** dict{'releve':<TRelevesOccurrence>, 'cruved': Cruved}

**DELETE** /occtax/releve/ (int: id\_releve)

Delete one releve and its associated occurrences and counting

**Params** int id\_releve ID of the releve to delete

**GET** /users/menu\_from\_code/ (string: code\_liste)

Retourne la liste des roles associés à une liste (identifiée par son code)

**Parameters**

- **code\_liste** (*string*) – the code of user list (utilisateurs.t\_lists)

#### Query Parameters

- **nom\_complet** (*str*) – begining of complet name of the role

**GET** /admin/static/ (**path:** *filename*)

Function used internally to send static files from the static folder to the browser.

New in version 0.5.

**GET** /users/menu/ (**int:** *id\_menu*)

Retourne la liste des roles associés à un menu

#### Parameters

- **id\_menu** (*int*) – the id of user list (utilisateurs.bib\_list)

#### Query Parameters

- **nom\_complet** (*str*) – begining of complet name of the role

**GET** /users/role/ (**int:** *id\_role*)

Get role detail

#### Parameters

- **id\_role** (*int*) – the id user

**DELETE** /meta/acquisition\_framework/ (**int:** *af\_id*)

Delete an acquisition framework .. :quickref: Metadata;

**POST** /meta/acquisition\_framework/ (**int:** *id\_acquisition\_framework*)

Post one AcquisitionFramework data for update acquisition\_framework .. :quickref: Metadata;

**GET** /meta/acquisition\_framework/ (*id\_acquisition\_framework*)

Get one AF with nomenclatures .. :quickref: Metadata;

#### Parameters

- **id\_acquisition\_framework** – the id\_acquisition\_framework
- **type** – int

**Returns** dict<TAcquisitionFramework>

**GET** /meta/dataset/ (**int:** *id\_dataset*)

Get one dataset

#### Parameters

- **id\_dataset** – the id\_dataset
- **type** – int

**Returns** dict<TDataset>

**DELETE** /meta/dataset/ (**int:** *ds\_id*)

Delete a dataset

**PATCH** /meta/dataset/ (**int:** *id\_dataset*)

Post one Dataset data for update dataset .. :quickref: Metadata;

**POST** /meta/dataset/ (**int:** *id\_dataset*)

Post one Dataset data for update dataset .. :quickref: Metadata;

## 6.3 Release

Pour sortir une nouvelle version de GeoNature :

- Faites les éventuelles Releases des dépendances (UsersHub, TaxHub, UsersHub-authentification-module, Nomenclature-api-module, GeoNature-atlas)
- Assurez-vous que les sous-modules git de GeoNature pointent sur les bonnes versions des dépendances
- Mettez à jour la version de GeoNature et éventuellement des dépendances dans `install/install_all/install_all.ini`, `config/settings.ini.sample`, `backend/requirements.txt`
- Complétez le fichier `docs/CHANGELOG.rst` (en comparant les branches <https://github.com/PnX-SI/GeoNature/compare/develop>) et dater la version à sortir
- Mettez à jour le fichier `VERSION`
- Remplissez le tableau de compatibilité des dépendances (`docs/versions-compatibility.rst`)
- Mergez la branche `develop` dans la branche `master`
- Faites la release (<https://github.com/PnX-SI/GeoNature/releases>) en la taguant `X.Y.Z` (sans `v` devant) et en copiant le contenu du Changelog
- Dans la branche `develop`, modifiez le fichier `VERSION` en `X.Y.Z.dev0` et pareil dans le fichier `docs/CHANGELOG.rst`

## 6.4 BDD

Mettre à jour le `ref_geo` à partir des données IGN scan express :

- Télécharger le dernier millésime : <http://professionnels.ign.fr/adminexpress>
- Intégrer le fichier Shape dans la BDD grâce à QGIS dans une table nommée `ref_geo.temp_fr_municipalities`
- Générer le SQL de création de la table : `pg_dump --table=ref_geo.temp_fr_municipalities --column-inserts -U <MON_USER> -h <MON_HOST> -d <MA_BASE> > fr_municipalities.sql`. Le fichier en sortie doit s'appeler `fr_municipalities.sql`
- Zipper le fichier SQL et le mettre sur le serveur <https://geonature.fr/data>
- Adapter le script `install_db.sh` pour récupérer le nouveau fichier zippé

## 6.5 Pratiques et règles de developpement

Afin de partager des règles communes de développement et faciliter l'intégration de nouveau code, veuillez lire les recommandations et bonnes pratiques recommandées pour contribuer au projet GeoNature.

### 6.5.1 Git

- Ne jamais faire de commit dans la branche `master` mais dans la branche `develop` ou idéalement dans une branche dédiée à la fonctionnalité (feature branch)
- Faire des pull request vers la branche `develop`
- Faire des `git pull` avant chaque développement et avant chaque commit
- Les messages de commits font référence à un ticket ou le ferment (`ref #12` ou `fixes #23`)
- Les messages des commits sont en anglais (dans la mesure du possible)

### 6.5.2 Backend

- Une fonction ou classe doit contenir une docstring en français. Les doctstrings doivent suivre le modèle NumPy/SciPy (voir <https://numpydoc.readthedocs.io/en/latest/format.html> et <https://realpython.com/documenting-python-code/#numpydocstrings-example>)
- Les commentaires dans le codes doivent être en anglais (ne pas s’empêcher de mettre un commentaire en français sur une partie du code complexe !)
- Assurez-vous d’avoir récupéré les dépendances dans les sous-modules git : `git submodule init` && `git submodule update` - Après un `git pull`, il faut mettre à jour les sous-modules : `git submodule update`
- Installer les `requirements-dev` (`cd backend && pip install -r requirements-dev.txt`) qui contiennent une série d’outils indispensables au développement dans GeoNature.
- Utiliser *blake* comme formateur de texte et activer l’auto-formatage dans son éditeur de texte (Tuto pour VsCode : <https://medium.com/@marcobelo/setting-up-python-black-on-visual-studio-code-5318eba4cd00>)
- Utiliser *pylint* comme formateur de code
- Respecter la norme PEP8 (assurée par les deux outils précédents)
- La longueur maximale pour une ligne de code est 100 caractères. Pour VsCode copier ces lignes le fichier `settings.json` :
- Respecter le snake case

```
"python.formatting.blackArgs": [  
    "--line-length",  
    "100"  
]
```

- Utiliser des doubles quotes pour les chaînes de caractères.

### 6.5.3 BDD

- Le noms des tables est préfixé par un “t” pour une table de contenu, de “bib” pour les tables de “dictionnaires” et de “cor” pour les tables de correspondance (relations N-N)
- Les schémas du coeur de GeoNature sont préfixés de “gn”
- Les schémas des protocoles ou modules GeoNature sont préfixés de “pr”
- Ne rien écrire dans le schéma `public`
- Ne pas répéter le nom des tables dans les noms des colonnes

### 6.5.4 Typescript

- Documenter les fonctions et classes grâce au JSDoc en français (<https://jsdoc.app/>)
- Les commentaires dans le codes doivent être en anglais (ne pas s’empêcher de mettre un commentaire en français sur une partie du code complexe !)
- Les messages renvoyés aux utilisateurs sont en français
- Installer les outils de développement : `npm install --only=dev`
- Utiliser *prettier* comme formateur de texte et activer l’autoformatage dans son éditeur de texte (VsCode dispose d’une extension Prettier : <https://github.com/prettier/prettier-vscode>)
- Utiliser *tslint* comme linter
- La longueur maximale pour une ligne de code est 100 caractères.

### 6.5.5 Angular

- Suivre les recommandations définies par le styleguide Angular: <https://angular.io/guide/styleguide>. C’est une ressources très fournie en cas de question sur les pratiques de développement (principe de séparation des principes, organisation des services et des composants)
- On privilégiera l’utilisation des reactive forms pour la construction des formulaires (<https://angular.io/guide/reactive-forms>). Ce sont des formulaires pilotés par le code, ce qui facilite la lisibilité et le contrôle de ceux-ci.
- Pour l’ensemble des composants cartographiques et des formulaires (taxonomie, nomenclatures...), il est conseillé d’utiliser les composants présents dans le module ‘GN2CommonModule’.

### 6.5.6 HTML

- La longueur maximale pour une ligne de code est 100 caractères.
- Lorsqu’il y a plus d’un attribut sur une balise, revenir à la ligne et aligner les attributs :

```
<button
  mat-raised-button
  color="primary"
  class="btn-action hard-shadow uppercase ml-3"
  data-toggle="collapse"
  data-target="#collapseAvance"
>
  Filtrer
</button>
```

- VsCode fournit un formateur de HTML par défaut (Dans les options de VsCode, tapez “wrap attributes” et sélectionner “force-expand-multiline”)

### 6.5.7 Style et ergonomie

- Boutons : On utilise les boutons d'Angular materials (<https://material.angular.io/components/button/overview>).
  - mat-raised-button pour les boutons contenant du texte
  - mat-fab ou mat-mini-fab pour les boutons d'actions avec seulement une icône
- Couleur des boutons :
  - Action : primary
  - Validation: vert (n'existant pas dans material: utiliser la classe *button-success*)
  - Suppression: warn
  - Navigation: basic
- Librairie d'icônes :
  - Utiliser la librairie material icons fournie avec le projet : <https://material.io/resources/icons/?style=baseline>  
(`<mat-icon> add </mat-icon>`)
- Formulaire :
  - Nous utilisons pour l'instant le style des formulaires Bootstrap (<https://getbootstrap.com/docs/4.0/components/forms/>). Une réflexion de migration vers les formulaires materials est en cours.
- Système de grille et responsive :
  - Utiliser le système de grille de bootstrap pour assurer le responsive design sur l'application. On ne vise pas l'utilisation sur mobile, mais à minima sur ordinateur portable de petite taille.

## 6.6 Développer et installer un gn\_module

GeoNature a été conçu pour fonctionner en briques modulaires.

Chaque protocole, répondant à une question scientifique, est amené à avoir son propre module GeoNature comportant son modèle de base de données (dans un schéma séparé), son API et son interface utilisateur.

Les modules développés s'appuieront sur le coeur de GeoNature qui est constitué d'un ensemble de briques réutilisables.

En base de données, le coeur de GeoNature est constitué de l'ensemble des référentiels (utilisateurs, taxonomique, nomenclatures géographique) et du schéma `gn_synthese` regroupant l'ensemble données saisies dans les différents protocoles (voir doc administrateur pour plus de détail sur le modèle de données).

L'API du coeur permet d'interroger les schémas de la base de données "coeur" de GeoNature. Une documentation complète de l'API est disponible dans la rubrique *API*.

Du côté interface utilisateur, GeoNature met à disposition un ensemble de composants Angular réutilisables (<http://pnx-si.github.io/GeoNature/frontend/modules/GN2CommonModule.html>), pour l'affichage des cartes, des formulaires etc...



### 6.6.1 Développer un gn\_module

Avant de développer un `gn_module`, assurez-vous d’avoir GeoNature bien installé sur votre machine (voir *Installation de GeoNature uniquement*).

Afin de pouvoir connecter ce module au “cœur”, il est impératif de suivre une arborescence prédéfinie par l’équipe GeoNature. Un template GitHub a été prévu à cet effet ([https://github.com/PnX-SI/gn\\_module\\_template](https://github.com/PnX-SI/gn_module_template)). Il est possible de créer un nouveau dépôt GitHub à partir de ce template, ou alors de copier/coller le contenu du dépôt dans un nouveau.

Cette arborescence implique de développer le module dans les technologies du cœur de GeoNature à savoir :

- Le backend est développé en Python grâce au framework Flask.
- Le frontend est développé grâce au framework Angular (voir la version actuelle du cœur)

GeoNature prévoit cependant l’intégration de module “externe” dont le frontend serait développé dans d’autres technologies. La gestion de l’intégration du module est à la charge du développeur.

- Le module se placera dans un dossier à part du dossier “GeoNature” et portera le suffixe “gn\_module”. Exemple : `gn_module_validation`
- La racine du module comportera les fichiers suivants :
  - `install_app.sh` : script bash d’installation des librairies python ou npm nécessaires au module
  - `install_env.sh` : script bash d’installation des paquets Linux
  - `requirements.txt` : liste des librairies python nécessaires au module
  - `manifest.toml` : fichier de description du module (nom, version du module, version de GeoNature compatible)
  - `conf_gn_module.toml` : fichier de configuration de l’application (livré en version sample)
  - `conf_schema.toml.py` : schéma ‘marshmallow’ (<https://marshmallow.readthedocs.io/en/latest/>) du fichier de configuration (permet de s’assurer la conformité des paramètres renseignés par l’utilisateur). Ce fichier doit contenir une classe `GnModuleSchemaConf` dans laquelle toutes les configurations sont synchronisées.
  - `install_gn_module.py` : script python lançant les commandes relatives à ‘installation du module (Base de données, ...)’. Ce fichier doit comprendre une fonction `gnmodule_install_app(gn_db, gn_app)` qui est utilisée pour installer le module (Voir l’[exemple du module CMR](#))
- La racine du module comportera les dossiers suivants :
  - `backend` : dossier comportant l’API du module utilisant un blueprint Flask
  - Le fichier `blueprint.py` comprend les routes du module (ou instancie les nouveaux blueprints du module)
  - Le fichier `models.py` comprend les modèles SQLAlchemy des tables du module.
  - `frontend` : le dossier `app` comprend les fichiers typescript du module, et le dossier `assets` l’ensemble des médias (images, son).
    - \* Le dossier `app` doit comprendre le “module Angular racine”, celui-ci doit impérativement s’appeler `gnModule.module.ts`
    - \* Le dossier `app` doit contenir un fichier `module.config.ts`. Ce fichier est automatiquement synchronisé avec le fichier de configuration du module `<GEONATURE_DIRECTORY>/external_modules/<nom_module>/conf_gn_module.toml` grâce à la commande `geonature update_module_configuration <nom_module>`. C’est à partir de ce fichier que toutes les configuration doivent pointer.

- \* A la racine du dossier `frontend`, on retrouve également un fichier `package.json` qui décrit l'ensemble des librairies JS nécessaires au module.

– `data` : ce dossier comprenant les scripts SQL d'installation du module

Le module est ensuite installable à la manière d'un plugin grâce à la commande `geonature install_gn_module` de la manière suivante :

```
# se placer dans le répertoire backend de GeoNature
cd <GEONATURE_DIRECTORY>/backend
# activer le virtualenv python
source venv/bin/activate
# lancer la commande d'installation
geonature install_gn_module <CHEMIN_ABSOLU_DU_MODULE> <URL_API>
# exemple geonature install_gn_module /home/moi/gn_module_validation /validation
```

### Bonnes pratiques Frontend

- Pour l'ensemble des composants cartographiques et des formulaires (taxonomie, nomenclatures...), il est conseillé d'utiliser les composants présents dans le module 'GN2CommonModule'.

Importez ce module dans le module racine de la manière suivante

```
import { GN2CommonModule } from '@geonature-common/GN2Common.module';
```

- Les librairies JS seront installées dans le dossier `node_modules` de GeoNature. (Il n'est pas nécessaire de réinstaller toutes les librairies déjà présentes dans GeoNature (Angular, Leaflet, ChartJS ...). Le `package.json` de GeoNature liste l'ensemble des librairies déjà installées et réutilisable dans le module.
- Les fichiers d'assets sont à ranger dans le dossier `assets` du frontend. Angular-cli impose cependant que tous les assets soient dans le répertoire mère de l'application (celui de GeoNature). Un lien symbolique est créé à l'installation du module pour faire entre le dossier d'assets du module et celui de Geonature.
- Utiliser `node_modules` présent dans GeoNature

Pour utiliser des librairies déjà installées dans GeoNature, utilisez la syntaxe suivante

```
import { TreeModule } from "@librairies/angular-tree-component";
```

L'alias `@librairies` pointe en effet vers le repertoire des `node_modules` de GeoNature

Pour les utiliser à l'intérieur du module, utiliser la syntaxe suivante

```

```

Exemple pour le module de validation

```

```

## 6.6.2 Installer un gn\_module

Renseignez l'éventuel fichier `config/settings.ini` du module.

Pour installer un module, rendez vous dans le dossier `backend` de GeoNature.

Activer ensuite le `virtualenv` pour rendre disponible les commandes GeoNature

```
source venv/bin/activate
```

Lancez ensuite la commande

```
geonature install_gn_module <mon_chemin_absolu_vers_le_module> <url_api>
```

Le premier paramètre est l'emplacement absolu du module sur votre machine et le 2ème le chemin derrière lequel on retrouvera les routes de l'API du module.

Exemple pour atteindre les routes du module de validation à l'adresse '<http://mon-geonature.fr/api/geonature/validation>'

Cette commande exécute les actions suivantes :

- Vérification de la conformité de la structure du module (présence des fichiers et dossiers obligatoires)
- Intégration du blueprint du module dans l'API de GeoNature
- Vérification de la conformité des paramètres utilisateurs
- Génération du routing Angular pour le frontend
- Re-build du frontend pour une mise en production

Complétez l'éventuelle configuration du module (`config/conf_gn_module.toml`) à partir des paramètres présents dans `config/conf_gn_module.toml.example` dont vous pouvez surcoucher les valeurs par défaut. Puis relancez la mise à jour de la configuration (depuis le répertoire `geonature/backend` et une fois dans le `venv` (`source venv/bin/activate`): `geonature update_module_configuration nom_du_module`)

## 6.7 Passer en mode développement

### 6.7.1 Récupération des sources

Si vous avez téléchargé GeoNature zippé (via la procédure d'installation globale `install_all.sh` ou en suivant la documentation d'installation standalone), il est nécessaire de rattacher votre répertoire au dépôt GitHub afin de pouvoir télécharger les dernières avancées du cœur en `git pull`. Pour cela, suivez les commandes suivantes en vous plaçant à la racine du répertoire de GeoNature.

```
--- Se créer un répertoire .git ---
mkdir .git
--- récupérer l'historique du dépôt ---
git clone --depth=2 --bare https://github.com/PnX-SI/GeoNature.git .git
--- initialiser un dépôt git à partir de l'historique téléchargé ---
git init
--- vérifier que le dépôt distant et le contenu local sont synchronisés ---
git pull
--- Reset sur HEAD pour mettre à jour les status ---
git reset HEAD
-> vous êtes à jour sur la branche master
--- Cloner les sous-modules pour récupérer les dépendances
```

(continues on next page)

(continued from previous page)

```
git submodule init
git submodule update
```

## 6.7.2 Configuration des URLs de développement

il est nécessaire de changer la configuration du fichier `config/geonature_config.toml` pour utiliser les adresses suivantes :

```
URL_APPLICATION = 'http://127.0.0.1:4200'
API_ENDPOINT = 'http://127.0.0.1:8000'
API_TAXHUB = 'http://127.0.0.1:5000/api'
```

Pour mettre à jour le fichier ``frontend/src/conf/app.config.ts`` et prendre en compte ces modifications, lancer les commandes suivantes :

## 6.7.3 Serveur frontend en développement

Lancer le serveur frontend via le virtualenv :

```
source ~/geonature/frontend/venv/bin/activate
geonature dev_front
```

Notez que vous pouvez aussi utiliser alternativement les commandes `npm` standards sans le virtualenv (consultez le fichier `frontend/package.json`).

## 6.7.4 API en développement

---

**Note:** Retrouvez plus de informations dans la section *Développement Backend* dédiée.

---

Dans un nouveau terminal, stopper le service `geonature` (`gunicorn`) et lancer le serveur backend :

```
sudo systemctl stop geonature
source ~/geonature/backend/venv/bin/activate
geonature dev_back
```

**Les serveurs seront accessibles via ces adresses (login `admin` et password `admin`) :**

- backend - 127.0.0.1:8000
- frontend - 127.0.0.1:4200

## 6.7.5 Autres extensions en développement

Il n'est pas forcément utile de passer toutes les extensions en mode développement. Pour plus d'informations, référez-vous aux documentations dédiées :

- <https://taxhub.readthedocs.io/fr/latest/installation.html#developpement>
- <https://usershub.readthedocs.io/fr/latest/>

Si toutefois TaxHub retourne une erreur 500 et ne répond pas sur l'URL <http://127.0.0.1:5000> alors vous pouvez avoir besoin de passer TaxHub en mode développement :

```
source ~/taxhub/venv/bin/activate
flask run
```

## 6.7.6 Debugger avec un navigateur

L'extension [Angular DevTools](#) permettra de debugger l'application dans la console du navigateur. Pour utiliser l'extension vous devez l'installer et passer obligatoirement en mode development.

Ouvrez le fichier `frontend/src/conf/app.config.ts` et modifiez la valeur `PROD_MOD` pour avoir :

```
"PROD_MOD": false
```

Si le mode production (`PROD_MOD`) est à `true`, alors vous n'êtes pas en mode production lors du lancement de la commande `npm run start`.

## 6.8 Développement Backend

### 6.8.1 Démarrage du serveur de dev backend

La commande `geonature` fournit la sous-commande `dev_back` pour lancer un serveur de test :

```
(venv)...$ geonature dev_back
```

### 6.8.2 Base de données avec Flask-SQLAlchemy

L'intégration de la base de données à GeoNature repose sur la bibliothèque [Flask-SQLAlchemy](#).

Celle-ci fournit un objet `db` à importer comme ceci : `from geonature.utils.env import db`

Cet objet permet d'accéder à la session SQLAlchemy ainsi :

```
from geonature.utils.env import db
obj = db.session.query(MyModel).get(1)
```

Mais il fournit une base déclarative `db.Model` permettant d'interroger encore plus simplement les modèles via leur attribut `query` :

```
from geonature.utils.env import db
class MyModel(db.Model):
    ...

obj = MyModel.query.get(1)
```

L'attribut `query` fournit [plusieurs fonctions](#) très utiles dont la fonction `get_or_404` :

```
obj = MyModel.query.get_or_404(1)
```

Ceci est typiquement la première ligne de toutes les routes travaillant sur une instance (route de type get/update/delete).

## Fonctions de filtrages

L'attribut `query` est une instance de la classe `flask_sqlalchemy.BaseQuery` qui peut être sur-chargée afin de définir de nouvelles fonctions de filtrage.

On pourra ainsi implémenter une fonction pour filtrer les objets auxquels l'utilisateur a accès, ou encore pour implémenter des filtres de recherche.

```
from flask import g
import sqlalchemy as sa
from flask_sqlalchemy import BaseQuery
from geonature.core.gn_permissions import login_required

class MyModelQuery(BaseQuery):
    def filter_by_scope(self, scope):
        if scope == 0:
            self = self.filter(sa.false())
        elif scope in (1, 2):
            filters = [ MyModel.owner==g.current_user ]
            if scope == 2 and g.current_user.id_organism is not None:
                filters.append(MyModel.owner.any(id_organism=g.current_user.id_
↪organism))
            self = self.filter(sa.or_(*filters))
        return self

class MyModel(db.Model):
    query_class = MyModelQuery

@login_required
def list_my_model():
    obj_list = MyModel.query.filter_by_scope(2).all()
```

## 6.8.3 Serialisation des modèles

### Avec Marshmallow

La bibliothèque [Marshmallow](#) fournit des outils de sérialisation et desérialisation.

Elle est intégrée à GeoNature par la bibliothèque [Flask-Marshmallow](#) qui fournit l'objet `ma` à importer comme ceci :

```
from geonature.utils.env import ma
```

Cette bibliothèque ajoute notamment une méthode `jsonify` aux schémas.

Les schémas Marshmallow peuvent être facilement créés à partir des modèles SQLAlchemy grâce à la bibliothèque [Marshmallow-SQLAlchemy](#).

```
from geonature.utils.env import ma

class MyModelSchema(ma.SQLAlchemyAutoSchema):
```

(continues on next page)

(continued from previous page)

```
class Meta:
    model = MyModel
    include_fk = True
```

La propriété `include_fk=True` concerne les champs de type `ForeignKey`, mais pas les relationships en elles-même. Pour ces dernières, il est nécessaire d'ajouter manuellement des champs `Nested` à son schéma :

```
class ParentModelSchema(ma.SQLAlchemyAutoSchema):
    class Meta:
        model = ParentModel
        include_fk = True

    childs = ma.Nested("ChildModelSchema", many=True)

class ChildModelSchema(ma.SQLAlchemyAutoSchema):
    class Meta:
        model = ChildModel
        include_fk = True

    parent = ma.Nested(ParentModelSchema)
```

Attention, la sérialisation d'un objet avec un tel schéma va provoquer une récursion infinie, le schéma parent incluant le schéma enfant, et le schéma enfant incluant le schéma parent.

Il est donc nécessaire de restreindre les champs à inclure avec l'argument `only` ou `exclude` lors de la création des schémas :

```
parent_schema = ParentModelSchema(only=['pk', 'childs.pk'])
```

L'utilisation de `only` est lourde puisqu'il faut re-spécifier tous les champs à sérialiser. On est alors tenté d'utiliser l'argument `exclude` :

```
parent_schema = ParentModelSchema(exclude=['childs.parent'])
```

Cependant, l'utilisation de `exclude` est hautement problématique !

En effet, l'ajout d'un nouveau champs `Nested` au schéma nécessiterait de le rajouter dans la liste des exclusions partout où le schéma est utilisé (que ça soit pour éviter une récursion infinie, d'alourdir une réponse JSON avec des données inutiles ou pour éviter un problème n+1 - voir section dédiée).

La bibliothèque `Utils-Flask-SQLAlchemy` fournit une classe utilitaire `SmartRelationshipsMixin` permettant de résoudre ces problématiques.

Elle permet d'exclure par défaut les champs `Nested`.

Pour demander la sérialisation d'un sous-schéma, il faut le spécifier avec `only`, mais sans nécessité de spécifier tous les champs basiques (non `Nested`).

```
from utils_flask_sqla.schema import SmartRelationshipsMixin

class ParentModelSchema(SmartRelationshipsMixin, ma.SQLAlchemyAutoSchema):
    class Meta:
        model = ParentModel
        include_fk = True

    childs = ma.Nested("ChildModelSchema", many=True)
```

(continues on next page)

(continued from previous page)

```
class ChildModelSchema(SmartRelationshipsMixin, ma.SQLAlchemyAutoSchema):
    class Meta:
        model = ChildModel
        include_fk = True

    parent = ma.Nested(ParentModelSchema)
```

### Avec le décorateur @serializable

---

**Note:** L'utilisation des schémas Marshmallow est probablement plus performante.

---

La bibliothèque maison [Utils-Flask-SQLAlchemy](#) fournit le décorateur @serializable qui ajoute une méthode as\_dict sur les modèles décorés :

```
from utils_flask_sqla.serializers import serializable

@serializable
class MyModel(db.Model):
    ...

obj = MyModel(...)
obj.as_dict()
```

La méthode as\_dict fournit les arguments fields et exclude permettant de spécifier les champs que l'on souhaite sérialiser.

Par défaut, seules les champs qui ne sont pas des relationship sont sérialisées (fonctionnalité similaire à celle fournit par SmartRelationshipsMixin pour Marshmallow).

Les relations que l'on souhaite voir sérialisées doivent être explicitement déclarées via l'argument fields.

L'argument fields supporte la « notation à point » permettant de préciser les champs d'un modèle en relation :

```
child.as_dict(fields=['parent.pk'])
```

Les tests unitaires fournissent un ensemble d'exemples d'usage du décorateur.

La fonction as\_dict prenait autrefois en argument les paramètres recursif et depth qui sont tous les deux obsolètes. Ces derniers ont différents problèmes :

- récursion infinie (contournée par un hack qui ne résoud pas tous les problèmes et qu'il serait souhaitable de voir disparaître)
- augmentation non prévue des données sérialisées lors de l'ajout d'une nouvelle relationship
- problème n+1 (voir section dédiée)



## Cas des modèles géographiques

La bibliothèque maison [Utils-Flask-SQLAlchemy-Geo](#) fournit des décorateurs supplémentaires pour la sérialisation des modèles contenant des champs géographiques.

- `utils_flask_sqla_geo.serializers.geoserializable`

Décorateur pour les modèles SQLA : Ajoute une méthode `as_geofeature` qui retourne un dictionnaire serialisable sous forme de Feature geojson.

Fichier définition modèle

```
from geonature.utils.env import DB
from utils_flask_sqla_geo.serializers import geoserializable

@geoserializable
class MyModel(DB.Model):
    __tablename__ = 'bla'
    ...
```

Fichier utilisation modèle

```
instance = DB.session.query(MyModel).get(1)
result = instance.as_geofeature()
```

- `utils_flask_sqla_geo.serializers.shapeserializable`

Décorateur pour les modèles SQLA :

- Ajoute une méthode `as_list` qui retourne l'objet sous forme de tableau (utilisé pour créer des shapefiles)
- Ajoute une méthode de classe `to_shape` qui crée des shapefiles à partir des données passées en paramètre

Fichier définition modèle

```
from geonature.utils.env import DB
from utils_flask_sqla_geo.serializers import shapeserializable

@shapeserializable
class MyModel(DB.Model):
    __tablename__ = 'bla'
    ...
```

Fichier utilisation modèle :

```
# utilisation de as_shape()
data = DB.session.query(MyShapeserializableClass).all()
MyShapeserializableClass.as_shape(
    geom_col='geom_4326',
    srid=4326,
    data=data,
    dir_path=str(ROOT_DIR / 'backend/static/shapefiles'),
    file_name=file_name,
)
```

- `utils_flask_sqla_geo.utilsgeometry.FionaShapeService`

Classe utilitaire pour créer des shapefiles.

La classe contient 3 méthodes de classe :

- `FionaShapeService.create_shapes_struct()` : crée la structure de 3 shapefiles (point, ligne, polygone) à partir des colonnes et de la géométrie passée en paramètre
- `FionaShapeService.create_feature()` : ajoute un enregistrement aux shapefiles
- `FionaShapeService.save_and_zip_shapefiles()` : sauvegarde et zip les shapefiles qui ont au moins un enregistrement:

```
data = DB.session.query(MySQLAModel).all()

for d in data:
    FionaShapeService.create_shapes_struct(
        db_cols=db_cols,
        srid=current_app.config['LOCAL_SRID'],
        dir_path=dir_path,
        file_name=file_name,
        col_mapping=current_app.config['SYNTHESE']['EXPORT_COLUMNS']
    )
    FionaShapeService.create_feature(row_as_dict, geom)
    FionaShapeService.save_and_zip_shapefiles()
```

## 6.8.4 Réponses

Voici quelques conseils sur l'envoi de réponse dans vos routes.

- Privilégier l'envoi du modèle sérialisé (vues de type create/update), ou d'une liste de modèles sérialisés (vues de type list), plutôt que des structures de données non conventionnelles.

```
def get_foo(pk):
    foo = Foo.query.get_or_404(pk)
    return jsonify(foo.as_dict(fields=...))

def get_foo(pk):
    foo = Foo.query.get_or_404(pk)
    return FooSchema(only=...).jsonify(foo)

def list_foo():
    q = Foo.query.filter(...)
    return jsonify([foo.as_dict(fields=...) for foo in q.all()])

def list_foo():
    q = Foo.query.filter(...)
    return FooSchema(only=...).jsonify(q.all(), many=True)
```

- Pour les listes vides, ne pas renvoyer le code d'erreur 404 mais une liste vide !

```
return jsonify([])
```

- Renvoyer une liste et sa longueur dans une structure de données non conventionnelle est strictement inutile, il est très simple d'accéder à la longueur de la liste en javascript via l'attribut `length`.
- Traitement des erreurs : utiliser [les exceptions prévues à cet effet](#) :

```
from werkzeug.exceptions import Forbidden, BadRequest, NotFound

def restricted_action(pk):
    if ...:
        raise Forbidden
```

- Penser à utiliser `get_or_404` plutôt que de lancer une exception `NotFound`
- Si l'utilisateur n'a pas le droit d'effectuer une action, utiliser l'exception `Forbidden` (code HTTP 403), et non l'exception `Unauthorized` (code HTTP 401), cette dernière étant réservée aux utilisateurs non authentifiés.
- Vérifier la validité des données fournies par l'utilisateur (`request.json` ou `request.args`) et lever une exception `BadRequest` si celles-ci ne sont pas valides (l'utilisateur ne doit pas être en mesure de déclencher une erreur 500 en fournissant une string plutôt qu'un int par exemple !).

\* Marshmallow peut servir à cela :

```
from marshmallow import Schema, fields, ValidationError
def my_route():
    class RequestSchema(Schema):
        value = fields.Float()
    try:
        data = RequestSchema().load(request.json)
    except ValidationError as error:
        raise BadRequest(error.messages)
```

\* Cela peut être fait avec `jsonschema` :

```
from jsonschema import validate as validate_json, ValidationError
def my_route():
    request_schema = {
        "type": "object",
        "properties": {
            "value": { "type": "number", },
        },
        "minProperties": 1,
        "additionalProperties": False,
    }
    try:
        validate_json(request.json, request_schema)
    except ValidationError as err:
        raise BadRequest(err.message)
```

- Pour les réponses vides (exemple : route de type delete), on pourra utiliser le code de retour 204 :

```
return '', 204
```

Lorsque par exemple une action est traitée mais aucun résultat n'est à renvoyer, inutile d'envoyer une réponse « OK ». C'est l'envoi d'une réponse HTTP avec un code égale à 400 ou supérieur qui entrainera le traitement d'une erreur côté frontend, plutôt que de se fonder sur le contenu d'une réponse non normalisée.

## Le décorateur `@json_resp`

Historiquement, beaucoup de vues sont décorées avec le décorateur `@json_resp`.

Celui-ci apparait aujourd'hui superflu par rapport à l'usage directement de la fonction `jsonify` fournie par Flask.

- `utils_flask_sqla_geo.serializers.json_resp`

Décorateur pour les routes : les données renvoyées par la route sont automatiquement serialisées en json (ou geojson selon la structure des données).

S'insère entre le décorateur de route flask et la signature de fonction

Fichier routes

```
from flask import Blueprint
from utils_flask_sqla.response import json_resp

blueprint = Blueprint(__name__)

@blueprint.route('/myview')
@json_resp
def my_view():
    return {'result': 'OK'}

@blueprint.route('/myerrview')
@json_resp
def my_err_view():
    return {'result': 'Not OK'}, 400
```

### 6.8.5 Problème « n+1 »

Le problème « n+1 » est un anti-pattern courant des routes de type « liste » (par exemple, récupération de la liste des cadres d'acquisition).

En effet, on souhaite par exemple afficher la liste des cadres d'acquisitions, et pour chacun d'entre eux, la liste des jeux de données :

```
af_list = AcquisitionFramework.query.all()

# with Marshmallow (and SmartRelationshipsMixin)
return AcquisitionFrameworkSchema(only=['datasets']).jsonify(af_list, many=True)

# with @serializable
return jsonify([af.as_dict(fields=['datasets']) for af in af_list])
```

Ainsi, lors de la sérialisation de chaque AF, on demande à sérialiser l'attribut `datasets`, qui est une `relationships` vers la liste des DS associés :

```
class AcquisitionFramework(db.Model)
    datasets = db.relationships(Dataset, uselist=True)
```

Sans précision, la [stratégie de chargement](#) de la relation `datasets` est `select`, c'est-à-dire que l'accès à l'attribut `datasets` d'un AF provoque une nouvelle requête `select` afin de récupérer la liste des DS concernés.

Ceci est généralement peu grave lorsque l'on manipule un unique objet, mais dans le cas d'une liste d'objet, cela génère 1+n requêtes SQL : une pour récupérer la liste des AF, puis une lors de la sérialisation de chaque AF pour récupérer les DS de ce dernier.

Cela devient alors un problème de performance notable !

Afin de résoudre ce problème, il nous faut joindre les DS à la requête de récupération des AF.

Pour cela, plusieurs solutions :

- Le spécifier dans la relationship :

```
class AcquisitionFramework(db.Model)
    datasets = db.relationships(Dataset, uselist=True, lazy='joined')
```

Cependant, cette stratégie s'appliquera (sauf contre-ordre) dans tous les cas, même lorsque les DS ne sont pas nécessaires, alourdissant potentiellement certaines requêtes qui n'en ont pas usage.

- Le spécifier au moment où la requête est effectuée :

```
from sqlalchemy.orm import joinedload

af_list = AcquisitionFramework.query.options(joinedload('datasets')).all()
```

Il est également possible de joindre les relations d'une relation, par exemple le créateur des jeux de données :

```
af_list = (
    AcquisitionFramework.query
    .options(
        joinedload('datasets').options(
            joinedload('creator'),
        ),
    )
    .all()
)
```

Afin d'être sûr d'avoir joint toutes les relations nécessaires, il est possible d'utiliser la stratégie `raise` par défaut, ce qui va provoquer le lancement d'une exception lors de l'accès à un attribut non pré-chargé, nous incitant à le joindre également :

```
from sqlalchemy.orm import raiseload, joinedload

af_list = (
    AcquisitionFramework.query
    .options(
        raiseload('*'),
        joinedload('datasets'),
    )
    .all()
)
```

Pour toutes les requêtes récupérant une liste d'objet, l'utilisation de la stratégie `raise` par défaut est grandement encouragée afin de ne pas tomber dans cet anti-pattern.

La méthode `as_dict` du décorateur `@serializable` accepte l'argument `unloaded='raise'` ou `unloaded='warn'` pour un résultat similaire (ou un simple warning).

L'utilisation de `raiseload`, appartenant au cœur de SQLAlchemy, reste à privilégier.

## 6.8.6 Export des données

TODO

## 6.8.7 Utilisation de la configuration

La configuration globale de l'application est contrôlée par le fichier `config/geonature_config.toml` qui contient un nombre limité de paramètres.

De nombreux paramètres sont néanmoins passés à l'application via un schéma Marshmallow (voir fichier `backend/geonature/utils/config_schema.py`).

Dans l'application flask, l'ensemble des paramètres de configuration sont utilisables via le dictionnaire `config`

```
from geonature.utils.config import config
MY_PARAMETER = config['MY_PARAMETER']
```

Chaque module GeoNature dispose de son propre fichier de configuration, (`module/config/cong_gn_module.toml`) contrôlé de la même manière par un schéma Marshmallow (`module/config/conf_schema.toml.py`).

Pour récupérer la configuration du module dans l'application Flask, il existe deux méthodes:

Dans le fichier `blueprint.py`

```
# Methode 1 :

from flask import current_app
MY_MODULE_PARAMETER = current_app.config['MY_MODULE_NAME']['MY_PARAMETER']
# ou MY_MODULE_NAME est le nom du module tel qu'il est défini dans le fichier
↳ ``manifest.toml`` et la table ``gn_commons.t_modules``

#Méthode 2 :
MY_MODULE_PARAMETER = blueprint.config['MY_MODULE_PARAMETER']
```

Il peut-être utile de récupérer l'ID du module GeoNature (notamment pour des questions droits). De la même manière que précédemment, à l'intérieur d'une route, on peut récupérer l'ID du module de la manière suivante

```
ID_MODULE = blueprint.config['ID_MODULE']
# ou
ID_MODULE = current_app.config['MODULE_NAME']['ID_MODULE']
```

Si on souhaite récupérer l'ID du module en dehors du contexte d'une route, il faut utiliser la méthode suivante

```
from geonature.utils.env import get_id_module
ID_MODULE = get_id_module(current_app, 'occtax')
```

## 6.8.8 Authentification et autorisations

### Restreindre une route aux utilisateurs connectés

Utiliser le décorateur `@login_required`:

```
from geonature.core.gn_permissions.decorators import login_required

@login_required
def my_protected_route():
    pass
```

### Connaitre l'utilisateur actuellement connecté

L'utilisateur courant est stocké dans l'espace de nom `g`:

```
from flask import g

print(g.current_user)
```

Il s'agit d'une instance de `pypnusershub.db.models.User`.

### Vérification des droits des utilisateurs

- `geonature.core.gn_permissions.decorators.check_cruved_scope`

Décorateur pour les routes : Vérifie les droits de l'utilisateur à effectuer une action sur la donnée et le redirige en cas de niveau insuffisant ou d'informations de session erronées

params :

- action <str:['C','R','U','V','E','D']> type d'action effectuée par la route (Create, Read, Update, Validate, Export, Delete)
- get\_role <bool:False> : si True, ajoute l'id utilisateur aux kwargs de la vue
- module\_code: <str:None> : Code du module (`gn_commons.t_modules`) sur lequel on veut récupérer le CRUVED. Si ce paramètre n'est pas passé, on vérifie le CRUVED de GeoNature

```
from flask import Blueprint
from geonature.core.gn_permissions.tools import get_or_fetch_user_cruved
from utils_flask_sqla.response import json_resp
from geonature.core.gn_permissions import decorators as permissions

blueprint = Blueprint(__name__)

@blueprint.route('/mysensibleview', methods=['GET'])
@permissions.check_cruved_scope(
    'R',
    True,
    module_code="OCCTAX"
)
@json_resp
def my_sensible_view(info_role):
    # Récupérer l'id de l'utilisateur qui demande la route
    id_role = info_role.id_role
```

(continues on next page)

(continued from previous page)

```

# Récupérer la portée autorisée à l'utilisateur pour l'action 'R' (read)
read_scope = info_role.value_filter
#récupérer le CRUVED complet de l'utilisateur courant
user_cruved = get_or_fetch_user_cruved(
    session=session,
    id_role=info_role.id_role,
    module_code=MY_MODULE_CODE,
)
return {'result': 'id_role = {}'.format(info_role.id_role)}

```

- `geonature.core.gn_permissions.tools.cruved_scope_for_user_in_module`
  - Fonction qui retourne le CRUVED d'un utilisateur pour un module et/ou un objet donné.
  - Si aucun CRUVED n'est défini pour le module, c'est celui de GeoNature qui est retourné, sinon 0.
  - Le CRUVED du module enfant surcharge toujours celui du module parent.
  - Le CRUVED sur les objets n'est pas hérité du module parent.

params :

- `id_role` <integer:None>
- `module_code` <str:None> : code du module sur lequel on veut avoir le CRUVED
- `object_code` <str:'ALL'> : code de l'objet sur lequel on veut avoir le CRUVED
- `get_id` <boolean: False> : retourne l'`id_filter` et non le `code_filter` si True

Valeur retournée : tuple

A l'indice 0 du tuple: <dict{str:str}> ou <dict{str:int}>, boolean) {'C': '1', 'R': '2', 'U': '1', 'V': '2', 'E': '3', 'D': '3'} ou {'C': 2, 'R': 3, 'U': 4, 'V': 1, 'E': 2, 'D': 2} si `get_id=True`

A l'indice 1 du tuple: un booléen spécifiant si le CRUVED est hérité depuis un module parent ou non.

```

from pypnusershub.db.tools import cruved_for_user_in_app

# récupérer le CRUVED de l'utilisateur 1 dans le module OCCTAX
cruved, herited = cruved_scope_for_user_in_module(
    id_role=1
    module_code='OCCTAX'
)
# récupérer le CRUVED de l'utilisateur 1 sur GeoNature
cruved, herited = cruved_scope_for_user_in_module(id_role=1)

```

## 6.9 Développement Frontend

### 6.9.1 Bonnes pratiques

- Chaque `gn_module` de GeoNature doit être un module Angular indépendant <https://angular.io/guide/ngmodule>.
- Ce `gn_module` peut s'appuyer sur une série de composants génériques intégrés dans le module `GN2CommonModule` et décrit ci-dessous



## 6.9.2 Les composants génériques

Un ensemble de composants décrits ci-dessous sont intégrés dans le coeur de GeoNature et permettent aux développeurs de simplifier la mise en place de formulaires ou de bloc cartographiques.

Voir la [DOCUMENTATION COMPLETE](#) sur les composants génériques.

NB : les composants de type “formulaire” (balise *input* ou *select*) partagent une logique commune et ont des `Inputs` et des `Outputs` communs, décrits ci-dessous. (voir <https://github.com/PnX-SI/GeoNature/blob/master/frontend/src/app/GN2CommonModule/form/genericForm.component.ts>).

Une documentation complète des composants génériques est [disponible ici](#)

NB : les composants de type “formulaire” (balise *input* ou *select*) partagent une logique commune et ont des `Inputs` et des `Outputs` communs, décrits ci-dessous. (voir <https://github.com/PnX-SI/GeoNature/blob/master/frontend/src/app/GN2CommonModule/form/genericForm.component.ts>).

- **Inputs**
  - L’input `parentFormControl` de type `FormControl` (<https://angular.io/api/forms/FormControl>) permet de contrôler la logique et les valeurs du formulaire depuis l’extérieur du composant. Cet input est **obligatoire** pour le fonctionnement du composant.
  - L’input `label` (string) permet d’afficher un label au dessus de l’input.
  - L’input `displayAll` (boolean, défaut = false) permet d’ajouter un item ‘tous’ sur les inputs de type `select` (Exemple : pour sélectionner tous les jeux de données de la liste)
  - L’input `multiSelect` (boolean, défaut = false) permet de passer les composants de type `select` en “multiselect” (sélection multiple sur une liste déroulante). Le `parentFormControl` devient par conséquent un tableau
  - L’input `searchBar` (boolean, défaut = false) permet de rajouter une barre de recherche sur les composants `multiselect`
  - L’input `disabled` (boolean) permet de rendre le composant non-saisissable
  - L’input `debounceTime` définit une durée en ms après laquelle les événements `onChange` et `onDelete` sont déclenchés suite à un changement d’un formulaire. (Par défaut à 0)

- **Outputs**

Plusieurs `Output` communs à ses composants permettent d’émettre des événements liés aux formulaires.

- `onChange` : événement émit à chaque fois qu’un changement est effectué sur le composant. Renvoie la valeur fraîche de l’input.
- `onDelete` : événement émit chaque fois que le champ du formulaire est supprimé. Renvoie un événement vide.

Ces composants peuvent être considérés comme des “dump components” ou “presentation components”, puisque que la logique de contrôle est déporté au composant parent qui l’accueil (<https://blog.angular-university.io/angular-2-smart-components-vs-presentation-components-whats-the-difference-when-to-use-each-and-why/>)

Un ensemble de composants permettant de simplifier l’affichage des cartographies Leaflet sont disponibles. Notamment un composant “map-list” permettant de connecter une carte avec une liste d’objets décrits en détail ci dessous.

## MapListComponent

Le composant MapList fournit une carte pouvant être synchronisée avec une liste. La liste, pouvant être spécifique à chaque module, elle n'est pas intégrée dans le composant et est laissée à la responsabilité du développeur. Le service MapListService offre cependant des fonctions permettant facilement de synchroniser les deux éléments.

Fonctionnalité et comportement offert par le composant et le service :

- Charger les données

Le service expose la fonction `getData(apiEndPoint, params?)` permettant de charger les données pour la carte et la liste. Cette fonction doit être utilisée dans le composant qui utilise le composant MapListComponent. Elle se charge de faire appel à l'API passée en paramètre et de rendre les données disponibles au service.

Le deuxième paramètre `params` est un tableau de paramètre(s) (facultatif). Il permet de filtrer les données sur n'importe quelle propriété du GeoJson, et également de gérer la pagination.

Exemple : afficher les 10 premiers relevés du `cd_nom` 212 :

```
mapListService.getData('occtax/releve',  
[{'param': 'limit', 'value': 10},  
{'param': 'cd_nom', 'value': 212}])
```

Exemple dans le module OccTax

L'API doit nécessairement renvoyer un objet comportant un GeoJson. La structure de l'objet doit être la suivante :

```
'total': nombre d'élément total,  
'total_filtered': nombre d'élément filtré,  
'page': numéro de page de la liste,  
'limit': limite d'élément renvoyé,  
'items': le GeoJson
```

Pour une liste simple sans pagination, seule la propriété 'items' est obligatoire.

- Rafraîchir les données

La fonction `refreshData(apiEndPoint, method, params?)` permet de rafraîchir les données en fonction de filtres personnalisés. Les paramètres `apiEndPoint` et `params` sont les mêmes que pour la fonction `getData`. Le paramètre `method` permet lui de choisir si on ajoute - `append` -, ou si on initialise (ou remplace) - `set` - un filtre.

Exemple 1 : Pour filtrer sur l'observateur 1, puis ajouter un filtre sur l'observateur 2 :

```
mapListService.refreshData('occtax/relevé', 'append', [{'param': 'observers',  
↪ 'value': 1}])
```

puis :

```
refreshData('occtax/relevé', 'append', [{'param': 'observers', 'value': 2}])
```

Exemple 2: pour filtrer sur le `cd_nom` 212, supprimer ce filtre et filtrer sur le `cd_nom` 214

```
mapListService.refreshData('occtax/relevé', 'set', [{'param': 'cd_nom', 'value': 1  
↪ 'value': 2}])
```

puis :

```
mapListService.refreshData('occtax/relevé', 'set, [{param': 'cd_nom', 'value': 2
↪ '}]})
```

- Gestion des événements :

- Au clic sur un marker de la carte, le service MapListService expose la propriété `selectedRow` qui est un tableau contenant l'id du marker sélectionné. Il est ainsi possible de surligner l'élément sélectionné dans la liste.
- Au clic sur une ligne du tableau, utiliser la fonction `MapListService.onRowSelected(id)` (id étant l'id utilisé dans le GeoJson) qui permet de zoomer sur le point sélectionné et de changer la couleur de celui-ci.

Le service contient également deux propriétés publiques `geoJsonData` (le geojson renvoyé par l'API) et `tableData` (le tableau de features du Geojson) qui sont respectivement passées à la carte et à la liste. Ces deux propriétés sont utilisables pour interagir (ajouter, supprimer) avec les données de la carte et de la liste.

- Selector : `pnx-map-list`

- Inputs :

**idName** Libellé de l'id du geojson (id\_releve, id)

Type: string

**height** Taille de l'affichage de la carte Leaflet

Type: string

Exemple d'utilisation avec une liste simple :

```
<pnx-map-list
  idName="id_releve_occtax"
  height="80vh">
</pnx-map-list>
<table>
  <tr ngFor="let row of mapListService.tableData" [ngClass]=" {'selected':
↪ mapListService.selectedRow[0]} == row.id ">
    <td (click)="mapListService.onRowSelect(row.id)"> Zoom on map </td>
    <td > {{row.observers}} </td>
    <td > {{row.date}} </td>
  </tr>
</table>
```

## Pnx-Municipalities

Suite à l'ajout d'un input "valueFieldName" pour "pnx-areas" et "pnx-municipalities" dans la version 2.9.0 de GeoNature, pour ceux qui utilisent le composant `pnx-municipalities`, l'idéal serait de traduire les données et les modèles et de passer du `code_insee` à `id_area`. La correspondance est immédiate (`area_code = code_insee`).

Cependant, pour garder la retrocompatibilité du composant `pnx-municipalities` veuillez ajouter :

### 6.9.3 Test end to end

Pour toute PR ou nouvelle fonctionnalité il est demandé d'écrire des tests. Pour les test e2e, la librairie Cypress est utilisé. Des exemples de tests peuvent être trouvé ici : <https://github.com/PnX-SI/GeoNature/tree/develop/frontend/cypress/integration> Les tests sont joués automatiquement sur Github-action lors de commit et PR sur la branch develop et master. Pour lancer les tests sur sa machine locale, utilisez la commande `npm run e2e && npm run e2e:coverage`. Celle-ci lance le serveur de frontend, joue les tests cypress et contrôle la couverture de test. Cette dernière est disponible dans le repertoire *frontend/coverage*.

- dans les templates : `[valueFieldName]=' 'area_code'` dans les templates
- dans les config (js, ts ou json) (attention à la casse) : `"value_field_name": "area_code"`
- dans le module Monitoring, ajouter aussi `"type_util": "area"`

## 6.10 Outils d'aide à la qualité du code

Des outils d'amélioration du code pour les développeurs peuvent être utilisés : flake8, pylint, pytest, coverage.

La documentation peut être générée avec Sphinx.

Les fichiers de configuration de ces outils se trouvent à la racine du projet :

- .pylint

Un fichier `.editorconfig` permettant de définir le comportement de votre éditeur de code est également disponible à la racine du projet.

### 6.10.1 Sphinx

Sphinx est un générateur de documentation.

Pour générer la documentation HTML, se placer dans le répertoire `docs` et modifier les fichiers `.rst`:

```
cd docs
make html
```

### 6.10.2 Pylint

Pylint fait la même chose que Flake8 mais il est plus complet, plus configurable mais aussi plus exigeant.

Pour inspecter le répertoire `geonature`:

```
cd backend
pylint geonature
```

### 6.10.3 tslint

tslint fait la même chose que pylint mais pour la partie frontend en typescript:

```
cd frontend  
npm run lint
```

### 6.10.4 Pytest

Pytest permet de mettre en place des tests fonctionnels et automatisés du code Python.

Les fichiers de test sont dans le répertoire `backend/geonature/tests`

```
pytest
```

### 6.10.5 Coverage

Coverage permet de donner une indication concernant la couverture du code par les tests.

```
pytest --cov=geonature --cov-report=html
```

Ceci génère un rapport html disponible dans `htmlcov/index.html`.



## COMPATIBILITE

Versions fournies et testées des dépendances

### 7.1 GeoNature 2.9.2

Application / Module	Version
TaxHub	1.9.4
UsersHub	2.2.2
Nomenclature-API	1.5.1
Authentication-API	1.5.9
Habref-API	0.3.0
Utils-Flask-SQLAlchemy	0.2.6
Utils-Flask-SQLAlchemy-Geo	0.2.2

### 7.2 GeoNature 2.9.1

Application / Module	Version
TaxHub	1.9.4
UsersHub	2.2.2
Nomenclature-API	1.5.1
Authentication-API	1.5.9
Habref-API	0.3.0
Utils-Flask-SQLAlchemy	0.2.6
Utils-Flask-SQLAlchemy-Geo	0.2.2

## 7.3 GeoNature 2.9.0

Application / Module	Version
TaxHub	1.9.3
UsersHub	2.2.2
Nomenclature-API	1.5.1
Authentication-API	1.5.9
Habref-API	0.3.0
Utils-Flask-SQLAlchemy	0.2.6
Utils-Flask-SQLAlchemy-Geo	0.2.2

## 7.4 GeoNature 2.8.0

Application / Module	Version
TaxHub	1.9.0
UsersHub	2.2.1
Nomenclature-API	1.4.4
Authentication-API	1.5.7
Habref-API	0.2.0
Utils-Flask-SQLAlchemy	0.2.4
Utils-Flask-SQLAlchemy-Geo	0.2.1

## 7.5 GeoNature 2.7.5

Application / Module	Version
TaxHub	1.8.1
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.7
Habref-API	0.1.6

## 7.6 GeoNature 2.7.4

Application / Module	Version
TaxHub	1.8.1
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.7
Habref-API	0.1.6



## 7.7 GeoNature 2.7.3

Application / Module	Version
TaxHub	1.8.1
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.7
Habref-API	0.1.6

## 7.8 GeoNature 2.7.2

Application / Module	Version
TaxHub	1.8.1
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.6
Habref-API	0.1.6

## 7.9 GeoNature 2.7.1

Application / Module	Version
TaxHub	1.8.1
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.6
Habref-API	0.1.6

## 7.10 GeoNature 2.7.0

Application / Module	Version
TaxHub	1.8.0
UsersHub	2.1.3
Nomenclature-API	1.3.8
Authentication-API	1.4.6
Habref-API	0.1.6

## 7.11 GeoNature 2.6.2

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.6
Authentication-API	1.4.5
Habref-API	0.1.5

## 7.12 GeoNature 2.6.1

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.6
Authentication-API	1.4.5
Habref-API	0.1.5

## 7.13 GeoNature 2.6.0

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.5
Authentication-API	1.4.5
Habref-API	0.1.5

## 7.14 GeoNature 2.5.5

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.15 GeoNature 2.5.4

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.16 GeoNature 2.5.3

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.17 GeoNature 2.5.2

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.18 GeoNature 2.5.1

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.19 GeoNature 2.5.0

Application / Module	Version
TaxHub	1.7.3
UsersHub	2.1.3
Nomenclature-API	1.3.4
Authentication-API	1.4.4
Habref-API	0.1.4

## 7.20 GeoNature 2.4.1

Application / Module	Version
TaxHub	1.7.0
UsersHub	2.1.2
Nomenclature-API	1.3.3
Authentication-API	1.4.3
Habref-API	0.1.3

## 7.21 GeoNature 2.4.0

Application / Module	Version
TaxHub	1.7.0
UsersHub	2.1.2
Nomenclature-API	1.3.3
Authentication-API	1.4.3
Habref-API	0.1.3

## 7.22 GeoNature 2.3.2

Application / Module	Version
TaxHub	1.6.5
UsersHub	2.1.1
Nomenclature-API	1.3.2
Authentication-API	1.4.3
Habref-API	0.1.2

## 7.23 GeoNature 2.3.1

Application / Module	Version
TaxHub	1.6.5
UsersHub	2.1.1
Nomenclature-API	1.3.2
Authentication-API	1.4.3
Habref-API	0.1.2

## 7.24 GeoNature 2.3.0

Application / Module	Version
TaxHub	1.6.3
UsersHub	2.1.0
Nomenclature-API	1.3.1
Authentication-API	1.4.3
Habref-API	0.1.2

## 7.25 GeoNature 2.2.0

Application / Module	Version
TaxHub	1.6.3
UsersHub	2.1.0
Nomenclature-API	1.3.0
Authentication-API	1.4.1
Occtax	1.0.0

## 7.26 GeoNature 2.1.2

Application / Module	Version
TaxHub	1.6.3
UsersHub	2.0.3
Nomenclature-API	1.2.6
Authentication-API	1.3.3
Occtax	1.0.0

## 7.27 GeoNature 2.1.1

Application / Module	Version
TaxHub	1.6.3
UsersHub	2.0.3
Nomenclature-API	1.2.6
Authentication-API	1.3.3
Occtax	1.0.0

## 7.28 GeoNature 2.1.0

Application / Module	Version
TaxHub	1.6.2
UsersHub	2.0.3
Nomenclature-API	1.2.5
Authentication-API	1.3.3
Occtax	1.0.0

## 7.29 GeoNature 2.0.0

Application / Module	Version
TaxHub	1.6.2
UsersHub	2.0.3
Nomenclature-API	1.2.3
Authentication-API	1.3.2
Occtax	1.0.0

## 7.30 GeoNature2 RC4.2

Application / Module	Version
TaxHub	1.6.2
UsersHub	2.0.3
Nomenclature-API	1.2.3
Authentication-API	1.3.2
Occtax	1.0.0

## 7.31 GeoNature2 RC4.1

Application / Module	Version
TaxHub	1.6.1
UsersHub	2.0.2
Nomenclature-API	1.2.3
Authentication-API	1.3.1
Occtax	1.0.0

## 7.32 GeoNature2 RC4

Application / Module	Version
TaxHub	1.6.0
UsersHub	2.0.0
Nomenclature-API	1.2.3
Authentication-API	1.3.1
Occtax	1.0.0

## 7.33 GeoNature2 RC3

Application / Module	Version
TaxHub	1.5.1
UsersHub	1.3.3
Nomenclature-API	1.2.2
Authentication-API	1.2.1
Occtax	1.0.0

## 7.34 GeoNature2 RC2

Application / Module	Version
TaxHub	1.5.0
UsersHub	1.3.2
Nomenclature-API	1.2.1
Authentication-API	1.2.0
Occtax	1.0.0

## 7.35 GeoNature2 RC1

Application / Module	Version
TaxHub	1.5.0
UsersHub	1.3.2
Nomenclature-API	1.2.1
Authentication-API	1.2.0
Occtax	1.0.0

## 7.36 GeoNature2 Beta5

Application / Module	Version
TaxHub	1.4.0
UsersHub	1.3.1
Nomenclature-API	1.1.0
Authentication-API	1.1.0
Occtax	1.0.0

## 7.37 GeoNature2 Beta4

Application / Module	Version
TaxHub	1.3.3
UsersHub	1.3.1
Nomenclature-API	1.0.0
Authentication-API	1.1.0
Occtax	1.0.0



## 8.1 Problèmes liés au frontend

### 8.1.1 Message d'erreur lors de la compilation du frontend

- **Probleme lié à Node-sass:**

```
at module.exports (/home/myuser/geonature2/frontend/node_modules/node-sass/lib/  
↪binding.js:15:13) at Object.<anonymous> (/home/myuser/geonature2/frontend/node_  
↪modules/node-sass/lib/index.js:14:35)
```

Lancer la commande : `npm rebuild node-sass --force`

- **Probleme de mémoire**

```
[26098:0x3d10640]    98298 ms: Scavenge 977.3 (1059.9) -> 962.2 (1059.9) MB, 18.5 / 0.  
↪0 ms allocation failure  
<--- JS stacktrace --->  
Cannot get stack trace in GC.  
FATAL ERROR: MarkingDeque::EnsureCommitted Allocation failed - process out of memory
```

Cela vient d'un manque de mémoire vive lors de l'exécution de la compilation frontend.

Executer les commandes suivantes pour libérer de l'espace mémoire (<https://stackoverflow.com/questions/26193654/node-js-catch-enomem-error-thrown-after-spawn>):

```
sudo fallocate -l 4G /swapfile Create a 4 gigabyte swapfile  
sudo chmod 600 /swapfile Secure the swapfile by restricting access to root  
sudo mkswap /swapfile Mark the file as a swap space  
sudo swapon /swapfile Enable the swap
```

- **Problème pour trouver le chemin de lancement du frontend**

```
Tried to find bootstrap code, but could not. Specify either statically analyzable_  
↪bootstrap code or pass in an entryModule to the plugins options.
```

Editez le fichier `/home/<my_user>/geonature/frontend/tsconfig.json` et renseignez les bons chemins vers le frontend de GeoNature :

```
"@angular/*": ["/home/<my_user>/geonature/frontend/node_modules/@angular/*"],  
"@geonature_common/*" : ["/home/<my_user>/geonature/frontend/src/app/GN2CommonModule/*  
↪"],  
"@geonature/*" : ["/home/<my_user>/geonature/frontend/src/app/*"],  
"@geonature_config/*" : ["/home/<my_user>/geonaturefrontend/src/conf/*"],
```

- **Problème d’affichage du Frontend**

Si vous rencontrez des problèmes de librairies Frontend qui n’ont pas bien été installées ou non accessibles, vous pouvez les réinstaller

- Supprimer le répertoire `frontend/node_modules`
- Réinstaller les dépendances du Frontend : Dans le répertoire `frontend`, lancez la commande `npm install`
- Reconstruire le Frontend : Dans le répertoire `frontend`, lancez la commande `npm run build`

## 8.2 Problèmes liés à la BDD

- Après un redémarrage de PostgreSQL (`sudo service postgresql restart`), celle-ci ne sera plus accessible par l’application et si vous tentez de vous connecter, vous aurez un message du type `LoginError`. Cela est lié au fait que lorsqu’on redémarre PostgreSQL, il faut aussi relancer les API de GeoNature, car cela génère des erreurs de transaction et de session entre l’API et PostgreSQL.

Donc à chaque `sudo service postgresql restart`, lancer un `sudo supervisorctl reload`

## 8.3 Problème de lancement l’API lié à supervisor

- Suite à une montée de version, si la commande `sudo supervisorctl reload` renvoie cette erreur:

```
error: <class 'socket.error'>, [Errno 2] No such file or directory: file: /usr/lib/
↳python2.7/socket.py line: 228
```

Il s’agit d’une erreur lié à supervisor, qui se charge de lancer les applications. Redémarrer le service puis relancer la commande:

```
sudo service supervisor start
sudo supervisorctl reload
```

Retester le fonctionnement de l’application grâce à la commande `ps -aux |grep gunicorn`, celle ci doit renvoyer plus d’une ligne

## AUTEURS

### 9.1 Parc national des Ecrins

- Gil Deluermoz
- Theo Lechemia
- Elie Bouttier
- Camille Monchicourt



### 9.2 Parc national des Cévennes

- Amandine Sahl
- Frédéric Fidon
- Joël Clément



### 9.3 Contributeurs

- Quang Pham (Parc national des Ecrins)
- Claire Lagaye (Parc national de la Vanoise)
- Xavier Arbez (Parc naturel régional du Pilat)
- Kevin Samuel
- Donovan Maillard (Flavia)
- Khanh-Chau Nguyen (Parc national des Ecrins)

- Etienne Trimaille (Parc national de Guadeloupe)
- Fred Cloitre (LPO Ardèche)
- Daniel Thonon (LPO Isère)
- Richard Vergely (Parc national des Ecrins)
- Robin Elzeard (Parc national des Ecrins)
- Sylvain Montagner (Parc naturel régional de Normandie-Maine)
- Patrick Kappeler (Natural Solutions)
- Julien Corny (Natural Solutions)
- Jean-Brieuc Lehébel-Péron (GRETIA)
- Jean-Baptiste Desbas (Picardie Nature)
- Romain Baghi (NEO)
- Sébastien Grimault
- Sébastien Pascal-Poher (Probesys)
- Jean-Pascal Milcent (CBNA)
- Florent Vollmer (Ouidou)
- Aboubakr Oudghiri (Ouidou)
- Matthieu Etourneau (Geofit)
- Florent Richard (Geofit)
- Judith Panijel (UMS Patrinat)
- Alain Laupin (UMS Patrinat)
- Ludovic Le Pontois (Parc national des Pyrénées)
- Vincent Prunet (Parc national du Mercantour)
- David Michallet (CEN 38)
- Rhandy Grard (Freelance / Natural Solutions)
- Maxime Toma (LPO France)
- Adrien Pajot (Natural Solutions)
- Maxime Vergez (Natural Solutions)
- Aurélie Jambon (Natural Solutions)
- Julien Graziani (Natural Solutions)
- Ophélie Da Silva (Natural Solutions)

## CHANGELOG

### 10.1 2.10.0 (unreleased)

#### Nouveautés

- Simplification du CRUVED minimum pour accéder à GeoNature, ne nécessitant plus d'avoir obligatoirement un CRUVED défini au niveau global de GeoNature (#1622)
- Ajout de la commande `geonature db status`

#### Notes de version

- Passage à angular 12 (développeur), exécutez les commandes suivantes :

```
# depuis le répertoire frontend
nvm use
npm install --legacy-peer-deps
# depuis le virtualenv
geonature generate_frontend_modules_route
geonature generate_frontend_tsconfig
geonature generate_frontend_tsconfig_app
```

- Modification du fichier SCSS du composant personnalisable introduction. Ce composant possède désormais un fichier SCSS avec une règle qui lui est propre. Il est donc nécessaire de :

- se placer dans le dossier du composant : `cd ~/geonature/frontend/src/custom/components/introduction/`
- copier le fichier `introduction.component.scss.sample` en `introduction.component.scss` avec : `cp introduction.component.scss.sample introduction.component.scss`
- modifier votre fichier `frontend/src/custom/components/introduction/introduction.component.ts` en :

\* remplaçant :

```
@Component ({
  selector: 'pnx-introduction',
  templateUrl: 'introduction.component.html'
})
```

\* par :

```
@Component ({
  selector: 'pnx-introduction',
  styleUrls: ['introduction.component.scss'],
  templateUrl: 'introduction.component.html'
})
```

- le changement sera pris en compte au prochain *build* du frontend de GeoNature.

## 10.2 2.9.2 (2022-02-15)

### Nouveautés

- Optimisation du nombre d'informations renvoyées par l'API pour les utilisateurs et les organismes
- Ajout d'une commande pour relancer le calcul de la sensibilité, utile en cas de modification du référentiel de sensibilité : `geonature sensitivity update-synthese`. Elle s'appuie sur la fonction `gn_synthese.update_sensitivity()`.
- Le niveau de diffusion dans la synthèse n'est plus calculé automatiquement à partir du niveau de sensibilité (#1711)
- Le niveau de sensibilité tient compte du comportement de l'occurrence (`OCC_COMPORTEMENT`), en plus du statut biologique (`STATUT_BIO`)
- Optimisation du recalcul de la sensibilité lors de la mise à jour de la synthèse (trigger `BEFORE` au lieu de `AFTER`)
- Ajout de tests unitaires sur les fonctions de calcul de la sensibilité

### Corrections

- Correction d'une régression sur la récupération de la liste des taxons (#1672)
- Correction de l'authentification au CAS de l'INPN
- Correction du calcul de la sensibilité (#1284) :
  - Gestion correcte de la présence de plusieurs règles avec et sans critère statut biologique
  - Utilisation de la règle la plus sensible quand plusieurs règles s'appliquent

### Notes de version

- La correction de la fonction de calcul de la sensibilité est suivie d'un recalcul automatique du niveau de sensibilité des données présentes dans la synthèse. Si vous ne souhaitez pas procéder à ce recalcul, ajoutez le paramètre `-x recompute-sensitivity=false` lors de la mise à jour de la base de données avec la commande `geonature db autoupgrade` (lancée automatiquement par le script `migration.sh`) :

```
(venv)$ geonature db autoupgrade -x recompute-sensitivity=false
```

- Le niveau de diffusion des données dans la synthèse est remis à `NULL` si celui-ci équivaut au niveau de sensibilité. Seuls les niveaux de diffusion qui diffèrent sont laissés intacts. Si vous souhaitez rectifier vous-mêmes vos niveaux de diffusion et ne pas les remettre à `NULL` quand ils sont équivalents au niveau de sensibilité, vous pouvez ajouter le paramètre `-x clear-diffusion-level=false` lors de la mise à jour de la base de données :

```
(venv)$ geonature db autoupgrade -x clear-diffusion-level=false
```

Si vous redescendez à l'état antérieur de votre base de données, les niveaux de diffusion seront restaurés à partir du niveau de sensibilité ; vous pouvez éviter ceci avec `-x restore-diffusion-level=false`.

## 10.3 2.9.1 (2022-01-27)

### Nouveautés

- Utilisation du paramètre `page` de Flask à la place du paramètre `maison_offset` pour la pagination des routes (rétro-compatible)
- Installation de TaxHub en version 1.9.4 (version corrective) par défaut
- Ajout du paramètre de configuration `CODE_APPLICATION` (par défaut GN) (#1635)

### Corrections

- Correction de l'URL de réinitialisation de mot passe envoyée par email (#1620)
- Correction d'un problème d'authentification avec le CAS
- Occtax : Correction des listes déroulantes masquées dans le bloc dénombrement, en rajoutant un scroll
- Correction de l'URL de l'API de TaxHub (slash final manquant) pour l'affichage des photos sur la fiche d'un profil de taxon
- Correction de la synchronisation des métadonnées depuis MTD
- Correction de la génération du token quand on utilise le CAS de l'INPN pour se connecter à GeoNature
- Correction des permissions trop restrictives d'accès aux données de la synthèse
- Correction de la pagination de la route `/color_taxon` en rajoutant un ordonnancement par `cd_nom` et `id_area` (utilisé par Occtax-mobile)
- Contournement d'un problème de redirection incorrecte par l'API de TaxHub lorsque celui-ci est mal configuré (#1438, #1616)

## 10.4 2.9.0 - Actias luna (2022-01-13)

### Profils de taxons

#### Nouveautés

- Construction automatique d'une fiche d'identité (profil) par taxon grâce aux observations validées présentes dans la base de données (altitude min/max, distribution spatiale, date de première/dernière observation, nombre de données valides, phénologie) (#917 par @DonovanMaillard, @lepontois, @Adrien-Pajot, @TheoLechemia, @bouttier, @amandine-sahl, @jpm-cbna)
  - Création d'un schéma `gn_profiles` dans la BDD contenant les tables, les vues, les fonctions et les paramètres de calcul des profils de taxons (<https://github.com/PnX-SI/GeoNature/blob/develop/data/core/profiles.sql>) (#1103)
  - Mise en place de l'API des profils de taxons (#1104)
  - Affichage des scores de chaque observation par rapport au profil du taxon dans la liste des observations du module Validation, ainsi que dans les fiches détails des observations dans les modules Synthèse et Validation (#1105)
  - Ajout de filtres des observations par score ou critère des profils de taxon dans le module Validation (#1105)
  - Ajout d'une alerte de contextualisation d'une observation par rapport au profil de taxon, lors de sa saisie dans le module Occtax

- Mise en place de paramètres pour activer ou non les profils de taxons, paramétrer leurs règles et définir le statut de validation pris en compte pour le calcul des profils (“Certain-très probable” et “Probable” par défaut)
- Documentation des profils de taxons et de leur paramétrage (<https://docs.geonature.fr/admin-manual.html#profils-de-taxons>)
- Suppression de la vue matérialisée `gn_synthese.vm_min_max_for_taxons` et de la fonction `gn_synthese.fct_calculate_min_max_for_taxon()` qui n’étaient pas utilisées
- [OCCTAX] Ajout d’un bouton permettant d’annuler la modification d’un taxon (#1508 par @jbrieuclp)
- [OCCTAX] Ajout de valeurs par défaut aux champs additionnels (`gn_commons.t_additional_fields.default_value`)
- [OCCTAX] Ajout d’un filtre avancé par `id_releve`
- [SYNTHESE] Ajout d’un filtre avancé par UUID d’observation (#973)
- Amélioration des listes déroulantes en passant à la librairie `ng-select2` pour les composants multiselects (#616 par @jbrieuclp)
- Gestion du référentiel des régions par Alembic (#1475)
- Ajout des anciennes régions (1970-2016), inactives par défaut, mais utiles pour les règles régionales de sensibilité
- Gestion du référentiel de sensibilité (règles nationales et régionales) par Alembic (#1576)
- Ajout d’une documentation sur le calcul de la sensibilité des observations (<https://docs.geonature.fr/admin-manual.html#gestion-de-la-sensibilite>, par @mvergez)
- [SYNTHESE] Amélioration de la fenêtre de limite d’affichage atteinte (#1520 par @jpm-cbna)
- [OCCHAB] Utilisation de tout Habref par défaut si aucune liste d’habitats n’est renseignée dans la configuration du module
- [METADONNEES] Attribuer des droits à un utilisateur sur un JDD si il a des droits sur son cadre d’acquisition
- Association automatique et paramétrable des jeux de données personnels auto-générés à des modules (Occtax par défaut) (#1555)
- Utilisation du C du CRUVED de l’utilisateur pour lister les jeux de données dans lesquels il peut ajouter des données dans les différents modules (et non plus le R du CRUVED sur GeoNature) (#659)

## Corrections

- [OCCTAX] Correction de l’enregistrement des dénombrements lors de l’enchaînement des relevés (#1479 par @jbrieuclp)
- [OCCTAX] Correction du filtre du champs “Habitat” par typologie d’habitat
- [ADMIN] Correction de l’affichage du module (#1427 par @jbrieuclp)
- [ADMIN] Sécurisation du module (#839)
- [VALIDATION] Corrections de la validation des observations (#1485 / #1529)
- [METADONNEES] Amélioration des performances (#1559)
- [METADONNEES] Correction de la suppression des JDD
- [METADONNEES] Correction de l’export PDF des JDD (#1544)
- [METADONNEES] Correction des permissions (#1528)
- [METADONNEES] Correction de la recherche avancée



- [SYNTHESE] Correction de la recherche sur les champs génériques de type nombre entier (#1519 par @jpm-cbna)
- [SYNTHESE] Correction des permissions
- [SYNTHESE] Correction du lien entre les filtres CA et JDD (#1530)
- [OCCHAB] Correction du chargement de la configuration, des fiches info et de la modification d'une station
- [METADONNEES] Améliorations des performances et des contrôles du formulaire des acteurs pour les JDD et les CA (par @joelclems)
- Correction de la redirection vers le formulaire de login en cas de cookie corrompu (#1550 par @antoinececcimnhn)
- Correction de la création de compte utilisateur (#1527)
- Mise à jour du module Habref-api-module pour corrections de certaines données d'Habref

### Développement

- Migration vers la librairie `gn-select2` pour les listes déroulantes des formulaires (#616 / #1285 par @jbriue-clp)
- Documentation de développement backend revue et complétée (#1559, <https://docs.geonature.fr/development.html#developpement-backend>)
- Amélioration de nombreuses routes et fonctions du backend
- Ajouts de tests automatisés du backend
- Mise en place d'une intégration continue pour exécuter automatiquement les tests backend et leur couverture de code avec GitHub Actions, à chaque commit ou pull request dans les branches `develop` ou `master` (#1568, <https://github.com/PnX-SI/GeoNature/actions>)
- [VALIDATION] Suppression des vues SQL et optimisation des routes
- Génération automatique et aléatoire du paramètre `SECRET_KEY`
- [SYNTHESE] Remplacement de `as_literal` par `json.loads`, plus performant (par @antoinececcimnhn)
- Possibilité de filter la route `synthese/taxa_distribution` par `id_source` (#1446 par @mvergez)
- Factorisation du composant `pnx-municipalities` avec le composant `pnx-areas`
- Ajout du composant `pnx-areas` dans `dynamic-form`
- Ajout d'un input `valueFieldName` aux composants `pnx-areas` et `pnx-municipalities`. Voir documentation (<https://github.com/PnX-SI/GeoNature/blob/develop/docs/development.rst#pnx-municipalities>)
- Mise à jour de nombreuses dépendances

### Notes de version

- La liste des JDD dans les modules de saisie (Occtax, Occhab, Monitoring et Import) se base désormais sur le C du CRUVED de l'utilisateur au niveau du module (ou du C du CRUVED de GeoNature si l'utilisateur n'a pas de CRUVED sur le module), au lieu du R de GeoNature jusqu'à présent. Vous devrez donc potentiellement adapter vos permissions à ce changement de comportement (#659)
- Si vous avez surcouché le paramètre de configuration `AREA_FILTERS` de la section [SYNTHESE], veuillez remplacer `id_type` par `type_code` (voir `ref_geo.bib_areas_types`)

```
AREA_FILTERS = [
    { label = "Communes", id_type = 25 }
]
```

devient

```
AREA_FILTERS = [
    { label = "Communes", type_code = "COM" }
]
```

- Si vous aviez modifié les colonnes de la liste des observations du module Validation en adaptant la vue `gn_validation.v_synthese_validation_forwebapp`, celle-ci a été supprimée et il suffit désormais d'indiquer les colonnes souhaitées dans la configuration du module. Voir documentation (<http://docs.geonature.fr/admin-manual.html#liste-des-champs-visibles>)
- Les nouvelles fonctionnalités liées aux profils de taxons nécessitent de rafraîchir des vues matérialisées à intervalles réguliers et donc de créer une tâche planifiée (cron). Voir documentation (<https://docs.geonature.fr/installation.html#taches-planifiees>)
- Les régions sont maintenant disponibles via des migrations Alembic. Si vous possédez déjà les régions, vous pouvez l'indiquer à Alembic :

```
geonature db upgrade ref_geo@head
geonature db stamp d02f4563bebe
```

- Le référentiel de sensibilité est désormais disponible via une migration Alembic. Celui-ci nécessite le référentiel des régions (branche Alembic `ref_geo_fr_regions`), ainsi que le référentiel des anciennes régions (branche Alembic `ref_geo_fr_regions_1970`) – l'installation de ces référentiels est automatique avec l'installation des règles de sensibilité.
  - Si vous possédez déjà le référentiel, vous pouvez l'indiquer à Alembic :

```
geonature db stamp 7dfd0a813f86
```

- Si vous avez installé GeoNature 2.8.X, le référentiel de sensibilité n'a pas été installé automatiquement. Vous pouvez l'installer manuellement :

```
geonature db upgrade ref_sensitivity_inpn@head
```

Par défaut, seule les règles nationales sont activées, vous laissant le soin d'activer vos règles locales en base vous-même. Vous pouvez également demander, lors de l'installation du référentiel, à activer (resp. désactiver) toutes les règles en ajout à la commande Alembic l'option `-x active=true` (resp. `-x active=false`).

- Si vous souhaitez surcoucher les paramètres par défaut de Unicorn (`app_name`, `timeout`...), depuis le passage à `systemd` dans la version 2.8.0, c'est désormais à faire dans un fichier `environ` à la racine du dossier de votre GeoNature (#1588, <https://docs.geonature.fr/admin-manual.html#parametres-unicorn>)
- Si vous les utilisez, mettez à jour les modules Import, Export et Monitoring dans leurs dernières versions compatibles avec le version 2.9.0 de GeoNature

## 10.5 2.8.1 (2021-10-17)

### Nouveautés

- Ajout de l'indication du département au formulaire des communes (#1480)
- Ajout des champs `group2inpn` et `regne` au formulaire des nomenclatures (#1481)

### Corrections

- Correction de la commande `geonature db autoupgrade`
- Mise-à-jour corrective de [UsersHub-authentication-module 1.5.7](#)

## 10.6 2.8.0 - Vaccinium myrtillus (2021-10-18)

### Gestion de la base de données avec Alembic

Avant de mettre à jour GeoNature, vérifiez que les modules que vous utilisez disposent d'une version compatible avec la 2.8.0, suite au passage à la version 3 de Marshmallow.

#### Nouveautés

- Support de Debian 11 / Python 3.9
- Passage de supervisor à systemd
- Gestion de la base de données et de ses évolutions avec Alembic (#880)
- Mise à jour de la procédure d'installation afin d'utiliser Alembic (#880)
- Révision et réorganisation des scripts et de la documentation d'installation
- Passage à la version 3 de Marshmallow (#1451)
- Suppression du paramètre `ID_APP`, celui-ci est automatiquement déterminé à partir de la base de données et du code de l'application
- Ajout d'un index sur le champs `ref_geo.l_areas.id_area`
- Mise à jour des dépendances
  - TaxHub 1.9.0
  - UsersHub-authentification-module 1.5.6
  - Nomenclature-api-module 1.4.4
  - Habref-api-module 0.2.0
  - Utils-Flask-SQLAlchemy 0.2.4
  - Utils-Flask-SQLAlchemy-Geo 0.2.1

#### Corrections

- Corrections et améliorations des formulaires dynamiques et des champs additionnels
- Correction de l'envoi d'email lors de la récupération du mot de passe (#1471)
- Occtax : Correction du focus sur le champs "taxon" quand on enchaîne les taxons (#1462)
- Occtax : Correction du formulaire de modification quand le relevé est une ligne ou un polygone (#1461)
- Occtax : Correction de la conservation de la date quand on enchaîne les relevés (#1442)
- Occtax : Correction du paramètre d'export des champs additionnels (#1440)
- Synthèse : correction de la recherche par jeu de données (#1494)
- Correction de l'affichage des longues listes déroulantes dans les champs additionnels (#1442)
- Mise à jour de la table `cor_area_synthese` lors de l'ajout de nouvelles zones via un trigger sur la table `l_areas` (#1433)
- Correction de l'export PDF des fiches de métadonnées (#1449)
- Jeux de données : correction de l'affichage des imports sources
- Correction de la configuration Apache et de la gestion par flask d'un GeoNature accessible sur un préfix (e.g. `/geonature`) (#1463)
- Correction de la commande `install_packaged_gn_module`

- Correction des champs additionnels de type boutons radios (#1464 et #1472)
- Occtax : Correction du contrôle des heures quand on est sur 2 mois distincts (#1468)
- Suppression de nombreux identifiants en dur dans les scripts SQL de création de la BDD
- Correction du trigger d'Occtax vers la Synthèse pour le champs Comportement (#1469)
- Correction des fonctions `get_default_nomenclature_value`
- Correction du composant `multiselect` (#1488)
- Correction du script `migrate.sh` pour récupérer le fichier `custom.scss` depuis son nouvel emplacement (#1430)
- Correction du paramètre `EXPORT_OBSERVERS_COL`
- Métadonnées : Suppression en cascade sur les tables `gn_meta.cor_dataset_territory` et `gn_meta.cor_dataset_protocol` (#1452)
- Correction de la commande `install_packaged_gn_module` : rechargement des entry points après installation avec `pip` d'un module packagé
- Correction d'un bug lors de l'ajout d'un cadre d'acquisition

### Développement

- Mise à jour de plusieurs dépendances
- Packaging des modules fournis avec GeoNature
- L'utilisateur connecté est maintenant accessible via `g.current_user`
- Nettoyage et refactoring divers

### Notes de version

- Mettre à jour [UsersHub en version 2.2.1](#) et [TaxHub en version 1.9.0](#) (si vous les utilisez) **en sautant leur étape de passage à Alembic** (car la mise à jour de GeoNature se charge désormais de mettre à jour aussi les schémas `taxonomie` et `utilisateurs`)
- Suppression de `supervisor` :
  - Stopper GeoNature : `sudo supervisorctl stop geonature2`
  - Supprimer le fichier de configuration `supervisor` de GeoNature : `sudo rm /etc/supervisor/conf.d/geonature-service.conf`
  - Si `supervisor` n'est plus utilisé par aucun service (répertoire `/etc/supervisor/conf.d/` vide), il peut être désinstallé (`sudo apt remove supervisor`)
- Suivre la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Passage à `systemd` :
  - Copier le fichier `install/assets/geonature.service` dans `/etc/systemd/system/`
  - Éditer `/etc/systemd/system/geonature.service` et remplacer les variables `${USER}` (votre utilisateur linux courant) et `${BASE_DIR}` (chemin absolu du répertoire de GeoNature) par les valeurs appropriées
  - Lancer la commande `sudo systemctl daemon-reload`
  - Pour démarrer GeoNature : `sudo systemctl start geonature`
  - Pour lancer GeoNature automatiquement au démarrage du serveur : `sudo systemctl enable geonature`

- Correction de la configuration Apache : si vous servez GeoNature sur un préfixe (typiquement /geonature/api), assurez vous que ce préfixe figure bien également à la fin des directives ProxyPass et ProxyPassReverse comme dans l'exemple suivant :

```
<Location /geonature/api>
  ProxyPass http://127.0.0.1:8000/geonature/api
  ProxyPassReverse http://127.0.0.1:8000/geonature/api
</Location>
```

Si vous servez GeoNature sur un sous-domaine, vérifiez ou modifiez la configuration Apache :

```
<Location /api>
  ProxyPass http://127.0.0.1:8000/api
  ProxyPassReverse http://127.0.0.1:8000/api
</Location>
```

Pensez à recharger Apache si vous êtes amené à en changer la configuration : `sudo systemctl reload apache2`

- Passage à Alembic :
  - S'assurer d'avoir une base de données de GeoNature en version 2.7.5
  - Si vous avez UsersHub installé, ajoutez dans votre configuration GeoNature la section suivante (en adaptant le chemin) :

```
[ALEMBIC]
VERSION_LOCATIONS = '/path/to/usershub/app/migrations/versions'
```

- Entrer dans le virtualenv afin d'avoir la commande geonature disponible : `source backend/venv/bin/activate`
- Exécuter les commandes suivantes afin d'indiquer à Alembic l'état de votre base de données :

```
geonature db stamp f06cc80cc8ba # GeoNature 2.7.5
geonature db stamp 0dfdbfbccd63 # référentiel géographique des communes
geonature db stamp 3fdaa1805575 # référentiel géographique des départements
geonature db stamp 586613e2faeb # référentiel géographique des mailles 1x1
geonature db stamp 7d6e98441e4c # référentiel géographique des mailles 5x5
geonature db stamp ede150d9afd9 # référentiel géographique des mailles 10x10
geonature db stamp 1715cf31a75d # MNT de l'IGN
```

- Si vous aviez déjà installé certains modules, vous devez l'indiquer à Alembic :
  - \* Module *Occtax* : `geonature db stamp f57107d2d0ad`
  - \* Module *Occhab* : `geonature db stamp 2984569d5df6`
- Mettre sa base de données à jour avec Alembic : `geonature db autoupgrade`

Pour plus d'information sur l'utilisation d'Alembic, voir la [documentation administrateur de GeoNature](#).

## 10.7 2.7.5 (2021-07-28)

### Corrections

- Compatibilité avec Occtax-mobile 1.3. Possibilité d'ajouter la query string `fields` sur la route `meta/datasets` pour choisir les champs renvoyés par l'API

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires

## 10.8 2.7.4 (2021-07-23)

### Corrections

- Correction d'un import manquant entrainant un problème de compilation du frontend (#1424)

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires

## 10.9 2.7.3 (2021-07-22)

### Nouveautés

- Métadonnées : ajout des paramètres `CD_NOMENCLATURE_ROLE_TYPE_DS` et `CD_NOMENCLATURE_ROLE_TYPE_AF` pour limiter les rôles utilisables au niveau des jeux de données et des cadres d'acquisition (#1417)
- Ajout de la commande `mtd_sync` qui permet de synchroniser les métadonnées de toute une instance depuis le flux MTD du SINP

### Corrections

- Correction de l'affichage des jeux de données sur les fiches des cadres d'acquisition (#1410)
- Doc : Précision des OS supportés (Debian 10 uniquement en production)

### Développement

- Support des commandes Flask au niveau de la commande `geonature (run, db, routes, shell...)`
- Ajout des sous-modules en tant que dépendances
- Ajout d'une commande `install_packaged_gn_module`

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires

## 10.10 2.7.2 (2021-07-05)

### Corrections

- OCCTAX : correction de la vérification du CRUVED (#1413)
- OCCTAX : correction du dégrisement du formulaire au chargement de la fonctionnalité “mes lieux” (#1414)
- OCCTAX : Déplacement des champs additionnels pour les dénombrements avant les médias (#1409)
- Suppression des champs additionnels de type “taxonomy” qui n’étaient pas supportés

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires
- Exécuter le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.7.1to2.7.2.sql>)

## 10.11 2.7.1 (2021-07-02)

### Corrections

- Installation des dépendances javascript des modules lors de la migration de version de GeoNature (#1252)
- Installation de la version 1.8.1 de TaxHub par défaut à la place de la 1.8.0
- Intégration de la documentation permettant de mettre en place l’accès public à GeoNature

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires

## 10.12 2.7.0 - Androsace delphinensis (2021-06-30)

Nécessite la version 1.8.x de TaxHub.

### Nouveautés

- Compatible avec TaxHub 1.8.x qui inclut notamment la migration (optionnelle) vers Taxref version 14 et l’ajout de la BDC statuts
- Installation globale migrée de Taxref version 13 à 14
- Occtax : Possibilité d’ajouter des champs additionnels par JDD ou globaux au module et documentation liée (#1007)
- Occtax/Synthese : Ajout des champs additionnels dans les exports (#1114)
- Occtax/Synthese : Affichage des champs additionnels dans les fiches info
- Customisation : possibilité de changer le CSS sans rebuild l’application
- Admin : Création d’un backoffice d’administration des champs additionnels (#1007)
- Admin : Création d’une documentation d’administration des champs additionnels (#1007)
- Occtax : Possibilité de désactiver la recherche de taxon par liste (#1315)

- Occtax : Par défaut la recherche de taxon n'interroge pas une liste mais tout Taxref, si aucune liste de taxons n'a été spécifiée dans la configuration du module Occtax (voir notes de version) (#1315)
- Occtax/Metadonnées : possibilité d'associer une liste de taxons à un JDD (implémenté uniquement dans Occtax) (#1315)
- Occtax : Possibilité d'ajouter les infos sur les médias dans les exports (paramètre `ADD_MEDIA_IN_EXPORT`) (#1326)
- Occtax : Possibilité de paramétrer l'affichage des champs du composant MEDIA dans OCCTAX (paramètre `MEDIA_FIELDS_DETAILS` - #1287)
- Occtax : Possibilité de filtrer la liste des habitats du formulaire avec les nouveaux paramètres `ID_LIST_HABITAT` et `CD_TYPO_HABITAT`
- Occtax : Possibilité d'ouvrir le module avec un JDD pré-selectionné en passant le paramètre `id_dataset` dans l'URL (#1071)
- Accueil : Réorganisation des blocs (#1375)
- Accueil : Ajout d'un paramètre contrôlant la fréquence de MAJ du cache des statistiques de la page d'accueil (`STAT_BLOC_TTL`, par défaut 1h: 3600 secondes) (#1320)
- Amélioration des performances de récupération des modules et du CRUVED
- Monitoring : Ajout d'un trigger garantissant la cohérence entre `date_min` et `date_max` et historisation de la table `gn_monitoring.cor_visit_observer` (#1247)
- La page d'authentification affiche désormais le nom de l'application (`appName`) défini dans la configuration de GeoNature (#1277)
- Possibilité d'ouvrir l'accès à GeoNature sans authentification (voir documentation d'administration) (#1323)
- Métadonnées : Optimisation du temps de chargement des listes des CA et JDD (#1291)
- Métadonnées : Passage de la version 1.3.9 du standard SINP à la version 1.3.10 et ajout des champs liés dans les formulaires (#1291)
- Métadonnées : Révision du design la partie "Acteurs" du formulaire et mise à part de l'acteur de type "Contact principal", obligatoire dans le standard SINP
- Métadonnées : Ordonnancement des JDD par leur nom
- Métadonnées : Ajout de la suppression en cascade au niveau des tables des CA et des JDD
- Métadonnées : Ajout d'un message quand un CA ou JDD n'a pas d'acteur (#1404)
- Synthèse et validation : Ajout d'un filtre avancé `Possède des médias` (#1179, #1338, #1180)
- Synthèse : Affichage du contenu json du champs des données additionnelles, dans la fiche détail d'une observation
- Synthèse : Ajout de la possibilité d'afficher la colonne "Effectif" dans la liste des observations
- DynamicForm : enrichissement des formulaires dynamiques pour les médias, l'ajout de liens externes
- Ajout d'une contrainte d'unicité de la combinaison des champs `id_type` et `area_code` dans `ref_geo.l_areas` (#1270)
- Ajout d'une contrainte d'unicité du champs `type_code` de la table `ref_geo.bib_areas_types`
- Mise à jour des versions de nombreuses dépendances Python et Javascript
- Support du gestionnaire d'erreurs Sentry
- Compression des images



- Ajout d'un script permettant d'identifier les doublons dans `data/scripts/duplicates_deletion` (#1324)
- Validation : possibilité de passer des fonctions dans la liste des colonnes affichées (pour décoder une nomenclature)
- Validation : Les paramètres `LIST_COLUMNS_FRONTEND` et `COLUMNS_API_VALIDATION_WEB_APP` sont regroupés en un seul paramètre nommé `COLUMN_LIST`. Voir le fichier `contrib/gn_module_validation/config/conf_gn_module.toml.example`

### Corrections

- Occtax : Correction d'un bug sur le champs observateur lors de la modification d'un relevé (#1177)
- Occtax : Renseignement par défaut de l'utilisateur connecté à la création d'un relevé en mode "observers\_txt" (#1292)
- Occtax : Déplacement des boutons d'action à gauche dans la liste des taxons d'un relevé pour éviter qu'ils soient masqués quand les noms de taxon sont longs (#1299 et #1337)
- Occtax : Correction de la possibilité de modifier un relevé si U=1 (#1365)
- Occtax : Suppression du zoom quand on localise le relevé (#1317)
- Occtax : Correction du nombre de lignes affichées après une recherche
- Occtax : Correction de la suppression d'un habitat lors de la modification d'un relevé (#1296)
- Occtax : Correction des champs "Habitat" et "Lieu" quand on enchaîne des relevés (#1191)
- Occtax : Correction de l'enchaînement des saisies (#1300)
- Occtax : Correction de l'affichage des taxons quand le nom est long (#1299, #1337)
- Occtax : Correction de l'observateur par défaut en mode `observers_txt`
- Occtax : Correction des messages d'information multiples (#1367)
- Occtax : Correction de la mise à jour du "digitiser" lors d'une édition de relevé (#1392)
- Occtax : Correction du trigger alimentant les observateurs de la synthèse depuis Occtax (#1399)
- Métadonnées : Correction de la suppression d'un JDD sans données, depuis la liste des JDD (#1312)
- Métadonnées : Correction de la récupération des valeurs de nomenclature depuis MTD n'existant pas dans GeoNature (#1297)
- Authentification : Redirection vers la page login après une période d'inactivité (#1193)
- Résolution des problèmes de permission sur le fichier `gn_errors.log` (#1003)

### Développement

- Possibilité d'utiliser la commande `flask` (eg `flask shell`)
- Préparation de l'utilisation d'Alembic pour la gestion des migrations de la structure de la BDD (#880)
- Possibilité d'importer des modules packagés (#1272)
- Réorganisation des fichiers `requirements` et installation des branches `develop` des dépendances du fichier `requirements-dev.txt`
- Simplification de la gestion des erreurs
- Création de templates pour les configurations Apache de GeoNature, TaxHub et UsersHub, utilisés par le script `install_all.sh`
- Ajout du plugin `leaflet-image`

- Ajout d'un champs `type` dans la table `gn_commons.t_modules` pour gérer le polymorphisme, utilisé dans le module Monitoring
- Ajout des champs `meta_create_date` et `meta_update_date` dans la table `gn_commons.t_modules`
- Diverses améliorations mineures de l'architecture du code

### Notes de version

Si vous mettez à jour GeoNature :

- Mettez à jour TaxHub 1.8.x avant d'effectuer la mise à jour de GeoNature : <https://github.com/PnX-SI/TaxHub/releases>
- Si vous utilisez le module Monitoring, mettez le à jour en version 0.2.4 minimum avant de mettre à jour GeoNature
- Si vous n'aviez pas renseigné de valeur pour le paramètre `id_taxon_list` dans le fichier `contrib/occtax/config/conf_gn_module.toml` du module Occtax, la liste 100 n'est plus passée par défaut et le module va rechercher sur tout Taxref. Si vous souhaitez utiliser une liste de taxons dans la saisie Occtax, veuillez renseigner l'identifiant de votre liste dans la configuration du module
- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires
- Exécuter le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.6.2to2.7.0.sql>)
- Le script SQL de mise à jour va supprimer et recréer les vues `pr_occtax.v_export_occtax` et `gn_synthese.v_synthese_for_export` pour y intégrer les champs additionnels. Si vous aviez modifié ces vues, adaptez le script de mise à jour de GeoNature 2.6.2 à 2.7.0, ou répercuter vos modifications après la mise à jour, à appliquer aussi dans votre éventuelle surcouche des paramètres `default_columns_export` (dans `contrib/occtax/config/conf_gn_module.toml`) et `EXPORT_COLUMNS` (dans `config/geonature_config.toml`)
- Le fichier de customisation CSS a été déplacé de `frontend/src/custom/custom.scss` vers `frontend/src/assets/custom.css` pour pouvoir être modifié sans devoir rebuild l'application. Son déplacement est fait automatiquement lors de la mise à jour de GeoNature. Si vous avez customisé les styles dans ce fichier et notamment fait référence à d'autres fichiers, vérifiez ou adaptez leurs chemins
- Si vous aviez renseigné un des deux paramètres `LIST_COLUMNS_FRONTEND`, `COLUMNS_API_VALIDATION_WEB_APP` dans le module Validation, il est nécessaire de les remplacer par le nouveau paramètre `COLUMN_LIST`. Voir le fichier `contrib/gn_module_validation/config/conf_gn_module.toml.example`
- Modifier dans le fichier `/etc/supervisor/conf.d/geonature-service.conf`, remplacer `gn_errors.log` par `supervisor.log` dans la variable `stdout_logfile` :

```
sudo sed -i 's|(\(stdout_logfile = .*\) /gn_errors.log|\1/supervisor.log|' /etc/  
↪supervisor/conf.d/geonature-service.conf  
sudo supervisorctl reload
```

## 10.13 2.6.2 (2021-02-15)

### Corrections

- Metadonnées : correction d'un bug sur la fiche JDD si le module d'import n'est pas installé
- Metadonnées : correction de l'affichage de certains champs sur la fiche des cadres d'acquisition
- Metadonnées : la recherche rapide n'est plus sensible à la casse

## 10.14 2.6.1 (2021-02-11)

### Corrections

- Correction de la fonction `gn_synthese.fct_tri_cal_sensi_diff_level_on_each_statement()` non compatible avec PostgreSQL 10 (#1255)
- Synthèse : correction de l'affichage du filtre "statut de validation" (#1267)
- Permissions : correction de l'URL de redirection après l'édition des permissions (#1253)
- Précision de la documentation de mise à jour de GeoNature (#1251)
- Ajout du paramètre `DISPLAY_EMAIL_INFO_OBS` dans le fichier d'exemple de configuration (#1066 par @jbdesbas)
- Sécurité : suppression d'une route inutile
- Correction de l'URL de la doc sur la page d'accueil

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires
- Exécuter le script de mise à jour de la BDD du sous-module de nomenclature : <https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update1.3.5to1.3.6.sql>
- Exécuter ensuite le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.6.0to2.6.1.sql>)
- Suivez la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

## 10.15 2.6.0 - Saxifraga (2021-02-04)

Nécessite Debian 10, car cette nouvelle version nécessite PostgreSQL 10 minimum (qui n'est pas fourni par défaut avec Debian 9) pour les triggers déclenchés "on each statement", plus performants.

### Nouveautés

- Sensibilité : Ajout d'un trigger sur la synthèse déclenchant automatiquement le calcul de la sensibilité des observations et calculant ensuite leur niveau de diffusion (si celui-ci est NULL) en fonction de la sensibilité (#413 et #871)
- Ajout du format GeoPackage (GPKG) pour les exports SIG, plus simple, plus léger, plus performant et unique que le SHAPEFILE. Les exports au format SHP restent pour le moment utilisés par défaut (modifiable dans la configuration des modules Occtax, Occhab et Synthèse) (#898)

- Performances : Suppression du trigger le plus lourd calculant les couleurs des taxons par unités géographiques. Il est remplacé par une vue utilisant le nouveau paramètre `gn_commons.t_parameters.occtaxmobile_area_type`, définissant le code du type de zonage à utiliser pour les unités géographiques dans Occtax-mobile (Mailles de 5km par défaut) (#997)
- Performances : Amélioration du trigger de la Synthèse calculant les zonages d'une observation en ne faisant un `ST_Touches()` seulement si l'observation n'est pas un point et en le passant `on each statement` (#716)
- Métadonnées : Refonte de la liste des CA et JDD avec l'ajout d'informations et d'actions, ainsi qu'une recherche avancée (#889)
- Métadonnées : Révision des fiches info des CA et JDD avec l'ajout d'actions, du tableau des imports et du téléchargement des rapports d'UUID et de sensibilité (#889)
- Métadonnées: Ajout de la fonctionnalité de fermeture (dépot) au niveau du CA (qui ferme tous les JDD du CA), seulement si le CA a au moins un JDD. Désactivée par défaut via le paramètre `ENABLE_CLOSE_AF` (#889 par @alainlaupinmnhn)
- Métadonnées : Possibilité d'envoyer un email automatique au créateur et à l'utilisateur d'un CA quand celui-ci est fermé (#889)
- Métadonnées : Possibilité d'ajouter un titre spécifique aux exports PDF des CA quand ceux-ci sont fermés, par exemple pour en faire un certificat (#889)
- Métadonnées : Possibilité d'importer directement dans un JDD actif depuis le module Métadonnées, désactivé par défaut (#889)
- Métadonnées : Amélioration des possibilités de customisation des PDF des fiches de métadonnées
- Métadonnées : Amélioration des fiches détail des CA et JDD et ajout de la liste des imports dans les fiches des JDD (#889)
- Métadonnées : Ajout d'un spinner lors du chargement de la liste des métadonnées et parallélisation du calcul du nombre de données par JDD (#1231)
- Synthèse : Possibilité d'ouvrir le module avec un JDD préselectionné (`<URL_GeoNature>/#/synthese?id_dataset=2`) et ajout d'un lien direct depuis le module Métadonnées (#889)
- Synthèse : ajout de web service pour le calcul du nombre d'observations par un paramètre donné (JDD, module, observateur), et du calcul de la bounding-box par jeu de données
- Synthèse : ajout d'un filtre avancé `Possède médias`
- Exports au format SHP remplacés par défaut par le format GeoPackage (GPKG) plus simple, plus léger, plus performant et unique. Les exports SHP restent activables dans la configuration des modules (#898)
- Occtax : ajout du paramètre `DISPLAY_VERNACULAR_NAME` qui contrôle l'affichage du nom vernaculaire vs nom complet sur les interfaces (Defaut = true: affiche le nom vernaculaire)
- Validation : Préremplir l'email à l'observateur avec des informations paramétrables sur l'occurrence (date, nom du taxon, commune, médias) (#981)
- Validation : Possibilité de paramétrer les colonnes affichées dans la liste des observations (#980)
- Possibilité de customiser le logo principal (GeoNature par défaut) dans `frontend/src/custom/images/`
- Ajout d'un champs json `additional_data` dans la table `l_areas` (#1111)
- Complément des scripts de migration des données depuis GINCO (`data/scripts/import_ginco/`)
- Barre de navigation : Mention plus générique et générale des auteurs et contributeurs
- Redirection vers le formulaire d'authentification si on tente d'accéder à une page directement sans être authentifié et sans passer par le frontend (#1193)

- Connexion à MTD : possibilité de filtrer les JDD par instance, avec le paramètre `ID_INSTANCE_FILTER`, par exemple pour ne récupérer que les JDD de sa région (#1195)
- Connexion à MTD : récupération du créateur et des acteurs (#922, #1008 et #1196)
- Connexion à MTD : récupération du nouveau champs `statutDonneesSource` pour indiquer si le JDD est d'origine publique ou privée
- Création d'une commande GeoNature permettant de récupérer les JDD, CA et acteurs depuis le webservice MTD de l'INPN, en refactorisant les outils existants d'import depuis ce webservice
- Ajout de contraintes d'unicité sur certains champs des tables de métadonnées et de la table des sources (#1215)
- Création d'un script permettant de remplacer les règles de sensibilité nationales et régionales, par les règles départementales plus précises (`data/scripts/sensi/import_sensi_depobio.sh`), uniquement utilisé pour DEPOBIO pour le moment, en attendant de clarifier dans une prochaine release le fonctionnement que l'on retient par défaut dans GeoNature (#413)
- Création d'un script permettant d'importer les régions dans le référentiel géographique (`data/migrations/insert_reg.sh`)

### Corrections

- Occhab : Export SIG (GPKG ou SHP) corrigé (#898)
- Meilleur nettoyage des sessions enregistrées dans le navigateur (#1178)
- Correction des droits CRUVED et de leur héritage (#1170)
- Synthèse : Retour du bouton pour revenir à l'observation dans son module d'origine (Occtax par exemple) depuis la fiche info d'une observation (#1147)
- Synthèse : Suppression du message "Aucun historique de validation" quand une observation n'a pas encore de validation (#1147)
- Synthèse : Correction du CRUVED sur le R = 1 (ajout des JDD de l'utilisateur)
- Synthèse : Correction de l'export des statuts basé sur une recherche géographique (#1203)
- Occtax : Correction de l'erreur de chargement de l'observateur lors de la modification d'un relevé (#1177)
- Occtax : Suppression de l'obligation de remplir les champs "Déterminateur" et "Méthode de détermination"
- Métadonnées : Suppression du graphique de répartition des espèces dans les exports PDF car il était partiellement fonctionnel
- Synthèse : Fonction `import_row_from_table`, test sur `LOWER(tbl_name)`
- Redirection vers le formulaire d'authentification si l'on essaie d'accéder à une URL sans être authentifié et sans passer par le frontend (#1193)
- Script d'installation globale : prise en compte du paramètre `install_grid_layer` permettant d'intégrer ou non les mailles dans le `ref_geo` lors de l'installation initiale (#1133)
- Synthèse : Changement de la longueur du champs `reference_biblio` de la table `gn_synthese`. `synthese` (de 255 à 5000 caractères)
- Sensibilité : Corrections des contraintes NOT VALID (#1245)

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires
- Exécuter ensuite le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.5.5to2.6.0.sql>)

- Toutes les nouvelles données intégrées dans la Synthèse auront leur niveau de sensibilité et de diffusion calculés automatiquement. Vous pouvez ajouter ou désactiver des règles de sensibilité dans la table `gn_sensitivity.t_sensitivity_rules`
- Vous pouvez aussi exécuter le script qui va calculer automatiquement le niveau de sensibilité et de diffusion de toutes les données déjà présentes dans la Synthèse, éventuellement en l'adaptant à votre contexte : <https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.5.5to2.6.0-update-sensitivity.sql>
- Mettez à jour de la longueur du champs `gn_synthese.synthese.reference_biblio` à 5000 caractères. Exécutez la commande suivante dans la console : `sudo -u postgres psql -d geonature2db -c "UPDATE pg_attribute SET atttypmod = 5004 WHERE attrelid = 'gn_synthese.synthese'::regclass AND attname = 'reference_biblio';"`
- Exécutez le script de mise à jour de la BDD du sous-module de nomenclature : <https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update1.3.4to1.3.5.sql>
- Suivez la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Si vous utilisez Occtax-mobile, vous pouvez modifier la valeur du nouveau paramètre `gn_commons.t_parameters.occtaxmobile_area_type` pour lui indiquer le code du type de zonage que vous utilisez pour les unités géographiques (mailles de 5km par défaut)
- Si vous disposez du module d'import, vous devez le mettre à jour en version 1.1.1

## 10.16 2.5.5 (2020-11-19)

### Nouveautés

- Ajout d'un composant fil d'ariane (#1143)
- Ajout de la possibilité de désactiver les composants `pnx-taxa` et `pnx-areas` (#1142)
- Ajout de tests sur les routes pour assurer la compatibilité avec les applications mobiles

### Corrections

- Correction d'un bug de récupération du CRUVED sur les modules (#1146)
- Correction des validateurs sur les preuves d'existence (#1134)
- Correction de la récupération des dossiers dans `backend/static` dans le script `migrate.sh`
- Correction de l'affichage de l'utilisateur dans la navbar lorsqu'on est connecté via le CAS INPN

## 10.17 2.5.4 (2020-11-17)

### Nouveautés

- Ajout de scripts `sql` et `sh` de restauration des medias dans `data/medias` (#1148)
- Ajout d'un service pour pouvoir récupérer les informations sur l'utilisateur connecté

### Corrections

- Correction des médias qui sont actuellement tous supprimés automatiquement après 24h, et non pas seulement ceux orphelins (#1148)
- Correction des permissions sur les fiches info des relevés dans Occtax avec la désactivation du bouton de modification du relevé quand l'utilisateur n'en a pas les droits

## Notes de version

- Si vous aviez associé des médias à des observations dans Occtax ou autre et qu'ils ont été supprimés, vous pouvez les retrouver dans la table d'historisation des actions (`SELECT * FROM gn_commons.t_history_actions WHERE table_content->'id_media' IS NOT NULL AND operation_type = 'D'`)
- Pour restaurer les médias supprimés depuis la table `gn_commons.t_history_actions` vous pouvez :
  - exécuter le script SQL `data/medias/restore_medias.sql` qui va recréer les médias supprimés dans la table `gn_commons.t_medias`
  - exécuter le script BASH `data/medias/restore_medias.sh` (`bash /home/~whoami~/geonature/data/medias/restore_medias.sh` en sudo si besoin) qui va renommer des fichiers supprimés en supprimant le préfixe `deleted_`

## 10.18 2.5.3 (2020-11-04)

### Nouveautés

- Mise en place de l'héritage du CRUVED au niveau des objets des modules (#1028)
- Révision de l'export des observations de la Synthèse (noms plus lisibles, ajout des communes et d'informations taxonomiques, complément des champs existants (#755)
- Ajout d'un paramètre permettant d'ajouter un message personnalisé à la fin des emails (inscriptions, exports...) (#1050 par @jpm-cbna)
- Ajout d'une alerte de dépréciation sur les fonctions `utils-sqlalchemy` présentes dans GeoNature
- Ajout d'un widget de type "HTML" dans les formulaires dynamiques, permettant d'ajouter des informations dans un formulaire (#1043 et #1068 par @jpm-cbna)
- Ajout de la possibilité d'ajouter un texte d'aide sur les champs des formulaires dynamiques (#1065 par @jpm-cbna)
- Ajout de la possibilité de définir un min et un max au composant commun `date` (#1069 par @jpm-cbna)
- Ajout de la possibilité de définir le nombre de lignes du composant commun `textarea` (#1067 par @jpm-cbna)
- Ajout de la possibilité de contrôler par une expression régulière le contenu d'un champs de type `text` des formulaires dynamiques (#1073 par @FlorentRICHARD44)
- Ajout de la possibilité de masquer certains champs du composant `media` (#1072, #1078 et #1083 par @me-tourneau)
- Ajout d'un spinner sur les statistiques de la page d'accueil (#1086 par @jpm-cbna)
- Ajout d'un composant d'autocomplete multiselect `pnx-taxa` permettant de rechercher des taxons dans tout l'arbre taxonomique et de limiter la recherche à un rang
- Possibilité d'ajouter plusieurs cartes sur la même page à l'aide du composant `pnx-map`
- Homogénéisation du style du code et documentation des pratiques de développement

### Corrections

- Correction de l'affichage des noms des validateurs sur la liste dans le module validation (#1091 par @lpofrede)
- Corrections mineures de l'export des observations de la Synthèse (#1108)
- Synthèse : Correction du masquage de la recherche par arbre taxonomique (#1057 par @jpm-cbna)

- Ajout du champs `id_nomenclature_biogeo_status` dans la Synthèse (correspondance standard : statut biogéographique). La BDD est remplie avec la valeur par défaut de la table `gn_synthese.default_nomenclature_value` (valeur = non renseignée)
- Accueil : Correction de l’affichage du nom du module (#1087)
- Correction du trigger de mise à jour d’Occtax vers la Synthèse (champs `the_geom_local` non mis à jour) (#1117 par @jbrieuclp)
- Correction du paramètre stockant la version de Taxref, passé à 13.0 pour les nouvelles installations (#1097 par @RomainBaghi)
- Correction de l’affichage en double des markers dans Leaflet.draw (#1095 par @FlorentRICHARD44)
- Synthèse : Correction des filtres avancés par technique d’observation et méthode de détermination (#1110 par @jbrieuclp)
- Recréation du fichier de configuration à chaque installation (#1074 par @etot)
- Annulation de l’insertion du module lorsqu’une erreur est levée à l’installation d’un module

#### Notes de version

- Désormais les objets des modules (par exemple les objets ‘Permissions’ et ‘Nomenclatures’ du module ‘ADMIN’) héritent automatiquement des permissions définies au niveau du module parent et à défaut au niveau de GeoNature (#1028). Il s’agit d’une évolution de mise en cohérence puisque les modules héritaient déjà des permissions de GeoNature, mais pas leurs objets. Si vous avez défini des permissions particulières aux niveaux des objets, vérifier leur cohérence avec le nouveau fonctionnement. NB : si vous aviez mis des droits R=0 pour un groupe au module ‘ADMIN’, les utilisateurs de ce groupe ne pourront pas accéder aux sous-modules ‘permissions’ et ‘nomenclatures’.
- Exécuter ensuite le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.5.2to2.5.3.sql>). Attention, si vous avez customisé les vues des exports Occtax et Synthèse, elles seront supprimées et recrées automatiquement par le script SQL de mise à jour de la BDD de GeoNature pour intégrer leurs évolutions réalisées dans cette nouvelle version. Réviser éventuellement ces vues avant et/ou après la mise à jour.
- Suivez la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>).
- Les noms de colonnes de l’export de la Synthèse ont été entièrement revus dans la vue fournie par défaut (`gn_synthese.v_synthese_for_export`). Si vous aviez surcouché le paramètre `EXPORT_COLUMNS` dans le fichier `config/geonature_config.toml`, vérifiez les noms des colonnes.
- Vérifiez que la valeur du paramètre `taxref_version` dans la table `gn_commons.t_parameters` correspond bien à votre version actuelle de Taxref (11.0 ou 13.0).

## 10.19 2.5.2 (2020-10-13)

#### Corrections

- Occtax : correction du problème d’installation du module dans le fichier `schemas.py`
- Synthèse : correction de la fonctions SQL `gn_synthese.import_row_from_table` et répercussion dans le fichier `gn_synthese/process.py`

#### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires



- Exécuter ensuite le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.5.1to2.5.2.sql>)

## 10.20 2.5.1 (2020-10-06)

### Corrections

- Ajout d'un paramètre `DISPLAY_EMAIL_INFO_OBS` définissant si les adresses email des observateurs sont affichées ou non dans les fiches info des observations des modules Synthèse et Validation (#1066)
- Occtax : correction de l'affichage du champs "Technique de collecte Campanule" (#1059)
- Occtax : correction du fichier d'exemple de configuration `contrib/occtax/config/conf_gn_module.toml.example` (#1059)
- Occtax : paramètre `DISPLAY_SETTINGS_TOOLS` renommé `ENABLE_SETTINGS_TOOLS` et désactivé par défaut (#1060)
- Occtax : quand le paramètre `ENABLE_SETTINGS_TOOLS` est désactivé, remise en place du fonctionnement de l'outil "Enchaîner les relevés". Dans ce cas, quand on enchaîne les relevés, on conserve le JDD, les observateurs, les dates et heures d'un relevé à l'autre (#1060)
- Occtax : correction de l'observateur par défaut en mode `observers_as_txt`
- Verification des UUID : autoriser toutes les versions (#1063)

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires

## 10.21 2.5.0 - Manidae (2020-09-30)

Occtax v2 et médias

### Nouveautés

- Refonte de l'ergonomie et du fonctionnement du module de saisie Occtax (#758 et #860 par @jbrieuclp et @TheoLechemia)
  - Enregistrement continu au fur et à mesure de la saisie d'un relevé
  - Découpage en 2 onglets (Un pour le relevé et un onglet pour les taxons)
  - Amélioration de la liste des taxons saisis sur un relevé (#635 et #682)
  - Amélioration de la saisie au clavier
  - Zoom réalisé dans la liste des relevé conservé quand on saisit un nouveau relevé (#436 et #912)
  - Filtres conservés quand on revient à la liste des relevés (#772)
  - Possibilité de conserver les informations saisies entre 2 taxons ou relevés, désactivable avec le paramètre `DISPLAY_SETTINGS_TOOLS` (#692)
  - Correction de la mise à jour des dates de début et de fin (#977)
  - Affichage d'une alerte si on saisit 2 fois le même taxon sur un même relevé
  - Fiche d'information d'un relevé complétée et mise à jour

- Passage de la version 1.2.1 à la version 2.0.0 du standard Occurrences de taxon (dans les modules Occtax, Synthèse et Validation) (#516)
  - Ajout des champs “Comportement”, “NomLieu”, “Habitat”, “Méthode de regroupement”, “Type de regroupement” et “Profondeur”
  - Ajout du champs “Précision” dans Occtax et suppression de sa valeur par défaut à 100 m
  - Renommage du champs “Méthode d’observation” en “Technique d’observation”
  - Suppression du champs “Technique d’observation” actuel de la synthèse
  - Renommage du champs “Technique d’observation” actuel d’Occtax en “Technique de collecte Campanule”
  - Ajout et mise à jour de quelques nomenclatures
  - Ajout d’un document de suivi de l’implémentation du standard Occurrences de taxon dans GeoNature (`docs/implementation_gn_standard_occtax2.0.ods`) (#516)
- Passage de la version 1.3.9 à la version 1.3.10 du standard de Métadonnées. Mise à jour des nomenclatures “CA\_OBJECTIFS” et mise à jour des métadonnées existantes en conséquence (par @DonovanMaillard)
- Ajout d’un champs `additional_data` de type `jsonb` dans la table `gn_synthese.synthese`, en prévision de l’ajout des champs additionnels dans Occtax et Synthèse (#1007)
- Mise en place de la gestion transversale et générique des médias (images, audios, vidéos, PDF...) dans `gn_commons.t_medias` et le Dynamic-Form (#336) et implémentation dans le module Occtax (désactivables avec le paramètre `ENABLE_MEDIAS`) (#620 par @joelclems)
- Mise en place de miniatures et d’aperçus des médias, ainsi que de nombreux contrôles des fichiers et de leurs formats
- Affichage des médias dans les fiches d’information des modules de saisie, ainsi que dans les modules Synthèse et Validation
- Ajout de la fonctionnalité “Mes lieux” (`gn_commons.t_places`), permettant de stocker la géométrie de lieux individuels fréquemment utilisés, implémentée dans le module cartographique d’Occtax (désactivable avec le paramètre `ENABLE_MY_PLACES`) (#246 par @metourneau)
- Tri de l’ordre des modules dans le menu latéral par ordre alphabétique par défaut et possibilité de les ordonner avec le nouveau champs `gn_commons.t_modules.module_order` (#787 par @alainlaupinmnhn)
- Arrêt du support de l’installation packagée sur Debian 9 et Ubuntu 16 pour passer à Python version 3.6 et plus
- Prise en charge de PostGIS 3 et notamment l’installation de l’extension `postgis_raster` (#946 par @jpm-cbna)
- Création de compte : Envoi automatique d’un email à l’utilisateur quand son compte est validé. Nécessite la version 2.1.3 de UsersHub (#862 et #1035 par @jpm-cbna)

### Ajouts mineurs

- Homogénéisation des styles des boutons (#1026)
- Factorisation du code des fiches infos d’une observation dans les modules Synthèse et Validation (#1053)
- Métadonnées : Ajout d’un paramètre permettant de définir le nombre de CA affichés sur la page (100 par défaut)
- Métadonnées : Tri des CA et JDD par ordre alphabétique
- Métadonnées : Ajout d’un champs `id_digitiser` dans la table des CA et des JDD, utilisé en plus des acteurs pour le CRUVED des JDD (#921)
- Dynamic-Form : Ajout d’un composant “select” prenant une API en entrée (#1029)

- Dynamic-Form : Ajout de la possibilité d’afficher une définition d’un champs sous forme de tooltip
- CAS INPN : Redirection vers la page de connexion de GeoNature quand on se déconnecte
- Ajout d’une contrainte d’unicité sur `schema_name` et `table_name` sur la table `gn_commons_bib_tables_location_unique` (#962)
- Ajout d’une contrainte d’unicité sur `id_organism` et `parameter_name` dans la table `gn_commons.t_parameters` (#988)
- Ajout de la possibilité de filtrer le composant `dataset` du Dynamic-Form par `module_code` pour pouvoir choisir parmi les JDD associées à un module (#964)
- Mise à jour de `psycopg2` en version 2.8.5, `sqlalchemy` en 1.3.19, `marshmallow` en 2.15.6, `virtualenv` en 20.0.31 (par @jpm-cbna)
- Mises à jour de sécurité diverses
- Améliorations des scripts `install/install_db.sh` et `install/install_app.sh` (par @jpm-cbna)
- Ajout de l’autocomplétion des commandes `geonature` (#999 par @jpm-cbna)
- Suppression du fichier `backend/gunicorn_start.sh.sample`
- Amélioration du script `install/migration/migration.sh` en vérifiant la présence des dossiers optionnels avant de les copier
- Amélioration des fonctions `gn_synthese.import_json_row_format_insert_data` et `gn_synthese.import_json_row` pour prendre en charge la génération des `geojson` dans `PostGIS 3`
- Documentation administrateur : Précisions sur les labels, pictos et ordres des modules dans le menu de navigation latéral

## Corrections

- Module Validation : Affichage des commentaires du relevé et de l’observation (#978 et #854)
- Module Validation : Ne lister que les observations ayant un `UUID` et vérification de sa validité (#936)
- Module Validation : Correction et homogénéisation de l’affichage et du tri des observations par date (#971)
- Module Validation : Correction de l’affichage du statut de validation après mise à jour dans la liste des observations (#831)
- Module Validation : Correction de l’affichage du nom du validateur
- Module Validation : Amélioration des performances avec l’ajout d’un index sur le champs `uuid_attached_row` de la table `gn_commons.t_validations` (#923 par @jbdesbas)
- Suppression du trigger en double `tri_insert_synthese_cor_role_relevés_occtax` sur `pr_occtax.cor_role_relevés_occtax` (#762 par @jbrieucpl)
- Passage des requêtes d’export de la synthèse en `POST` plutôt qu’en `GET` (#883)
- Correction du traitement du paramètre `offset` de la route `synthese/color_taxon` utilisé par `Occtax-mobile` (#994)
- Correction et complément des scripts de migration de données depuis `GINCO v1` (`data/scripts/import_ginco/occtax.sql`)
- Import des utilisateurs depuis le CAS INPN : Activer les utilisateurs importés par défaut et récupérer leur email
- Calcul automatique de la sensibilité : Ajout de la récursivité dans la récupération des critères de sensibilité au niveau de la fonction `gn_sensitivity.get_id_nomenclature_sensitivity` (#284)
- Typo sur le terme “Preuve d’existence” (par @RomainBaghi)

## Notes de version

Si vous mettez à jour GeoNature :

- Nomenclatures : Commencer par exécuter le script SQL de mise à jour du schéma `ref_nomenclatures` de la BDD (<https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update1.3.3to1.3.4.sql>)
- Exécuter ensuite le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.4.1to2.5.0.sql>). Attention, si vous avez customisé les vues des exports Occtax et Synthèse, elles seront supprimées et recrées automatiquement par le script SQL de mise à jour de la BDD de GeoNature pour s'adapter aux évolutions du standard Occtax en version 2.0.0. Réviser éventuellement ces vues avant et/ou après la mise à jour. Le script SQL de mise à jour vérifiera aussi si vous avez d'autres vues (dans le module Export notamment) qui utilisent le champs `id_nomenclature_obs_technique` qui doit être renommé et l'indiquera dès le début de l'exécution du script, en l'arrêtant pour que vous puissiez modifier ou supprimer ces vues bloquant la mise à jour.
- Les colonnes de l'export de la Synthèse ont été partiellement revus dans la vue fournie par défaut (`gn_synthese.v_synthese_for_export`). Si vous aviez surcouché le paramètre `EXPORT_COLUMNS` dans le fichier `config/geonature_config.toml`, vérifiez les noms des colonnes.
- A partir la version 2.5.0 de GeoNature, la version 3.5 de Python n'est plus supportée. Seules les versions 3.6 et + le sont. Si vous êtes encore sur Debian 9 (fourni par défaut avec Python 3.5), veuillez suivre les instructions de mise à jour de Python sur cette version (<https://github.com/PnX-SI/GeoNature/blob/master/docs/installation-standalone.rst#python-37-sur-debian-9>). Il est cependant plutôt conseillé de passer sur Debian 10 pour rester à jour sur des versions maintenues
- Suivez la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- A noter, quelques changements dans les paramètres du module Occtax. Les paramètres d'affichage/masquage des champs du formulaire ont évolué ainsi :
  - `obs_meth` devient `obs_tech`
  - `obs_technique` devient `tech_collect`
- A noter aussi que cette version de GeoNature est compatible avec la version 1.1.0 minimum d'Occtax-mobile (du fait de la mise du standard Occurrence de taxons)

## 10.22 2.4.1 (2020-06-25)

### Nouveautés

- Occurrences sans géométrie précise : Ajout d'un champs `id_area_attachment` dans la table `gn_synthese.synthese` permettant d'associer une observation à un zonage dans le référentiel géographique (`ref_geo.l_areas.id_area`) (#845 et #867)
- Ajout d'un champs `geojson_4326` dans la table `ref_geo.l_areas` pour pouvoir afficher les zonages du référentiel géographique sur les cartes (#867)
- Ajout de l'import par défaut des départements de France métropole dans le référentiel géographique lors de l'installation de GeoNature (en plus des actuelles communes et grilles)
- Mise à jour des communes importées sur la version de février 2020 d'Admin express IGN pour les nouvelles installations

### Corrections

- Correction d'un bug d'affichage des fonds de carte WMTS de l'IGN, apparu dans la version 2.4.0 avec l'ajout du support des fonds WMS (#890)

- Gestion des exceptions de type `FileNotFoundException` lors de l'import des commandes d'un module

### Notes de version

Si vous mettez à jour GeoNature :

- Vous pouvez passer directement à cette version mais en suivant les notes des versions intermédiaires
- Exécuter le script SQL de mise à jour de la BDD de GeoNature : <https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.4.0to2.4.1.sql>
- Suivez la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Vous pouvez alors lancer le script d'insertion des départements de France métropole dans le référentiel géographique (optionnel) : [https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.4.0to2.4.1\\_insert\\_departements.sh](https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.4.0to2.4.1_insert_departements.sh). Vérifier le déroulement de l'import dans le fichier `var/log/insert_departements.log`

## 10.23 2.4.0 - Fiches de métadonnées (2020-06-22)

### Nouveautés

- Métadonnées : Ajout d'une fiche pour chaque jeu de données et cadres d'acquisition, incluant une carte de l'étendue des observations et un graphique de répartition des taxons par Groupe INPN (#846 par @FloVollmer)
- Métadonnées : Possibilité d'exporter les fiches des JDD et des CA en PDF, générés par le serveur avec WeasyPrint. Logo et entêtes modifiables dans le dossier `backend/static/images/` (#882 par @FloVollmer)
- Métadonnées : Implémentation du CRUVED sur la liste des CA et JDD (#911)
- Métadonnées : Affichage de tous les CA des JDD pour lesquels l'utilisateur connecté a des droits (#908)
- Compatible avec TaxHub 1.7.0 qui inclut notamment la migration (optionnelle) vers Taxref version 13
- Installation globale migrée de Taxref version 11 à 13
- Synthèse et zonages : Ne pas inclure l'association aux zonages limitrophes d'une observation quand sa géométrie est égale à un zonage (maille, commune...) (#716 par @jbdesbas)
- Synthèse : Ajout de la possibilité d'activer la recherche par observateur à travers une liste, avec ajout des paramètres `SEARCH_OBSERVER_WITH_LIST` (False par défaut) et `ID_SEARCH_OBSERVER_LIST` (#834 par @jbrieuclp)
- Synthèse : Amélioration de la recherche des observateurs. Non prise en compte de l'ordre des noms saisis (#834 par @jbrieuclp)
- Synthèse : Ajout de filtres avancés (Saisie par basé sur `id_digitiser`, Commentaire du relevé et de l'occurrence, Déterminateur) (#834 par @jbrieuclp)
- Occtax : Création d'un trigger générique de calcul de l'altitude qui n'est exécuté que si l'altitude n'est pas postée (#848)
- Ajout d'une table `gn_commons.t_mobile_apps` permettant de lister les applications mobiles, l'URL de leur APK et d'une API pour interroger le contenu de cette table. Les fichiers des applications et leurs fichiers de configurations peuvent être chargés dans le dossier `backend/static/mobile` (#852)
- Ajout d'un offset et d'une limite sur la route de la couleur des taxons (utilisée uniquement par Occtax-mobile actuellement)

- Support des fonds de carte au format WMS (<https://leafletjs.com/reference-1.6.0.html#tilelayer-wms-l-tilelayer-wms>), (#890 par @jbdesbas)
- Ajout d'un champs texte `reference_biblio` dans la table `gn_synthese`
- Amélioration des performances du module de validation, en revoyant la vue `gn_commons.v_synthese_validation_forwebapp`, en revoyant les requêtes et en générant le GeoJSON au niveau de la BDD (#923)
- Ajout d'une fonction SQL d'insertion de données dans la `synthese` (et une fonction python associée)
- Compléments de la documentation (Permissions des utilisateurs, Occhab...)
- Ajout de scripts de migration des données de GINCO1 vers GeoNature (`data/scripts/import_ginco`)
- Trigger Occtax vers Synthèse : Amélioration du formatage des heures avec `date_trunc()` dans la fonction `pr_occtax.insert_in_synthese()` (#896 par @jbdesbas)
- Barre de navigation : Clarification de l'icône d'ouverture du menu, ajout d'un paramètre `LOGO_STRUCTURE_FILE` permettant de changer le nom du fichier du logo de l'application (#897 par @jbrieuclp)
- Médias : Amélioration des fonctions backend
- Mise à jour de jQuery en version 3.5.0
- Suppression de la table `gn_synthese.taxons_synthese_autocomplete` et du trigger sur la Synthèse qui la remplissait pour utiliser la vue matérialisée `taxonomie.vm_taxref_list_forautocomplete` listant les noms de recherche de tous les taxons de Taxref, entièrement revue dans TaxHub 1.7.0
- Monitoring : Correction du backend pour utiliser la nouvelle syntaxe de jointure des tables
- Ajout de fonctions SQL d'insertion de données dans la Synthèse (`gn_synthese.import_json_row()` et `gn_synthese.import_row_from_table()`) et de la fonction Python associée (`import_from_table(schema_name, table_name, field_name, value)`) pour l'API permettant de poster dans la Synthèse (#736). Utilisée par le module Monitoring.
- Ajout du plugin Leaflet.Deflate (#934 par @jpm-cbna)
- Connexion au CAS INPN : Association des JDD aux modules Occtax et Occhab (paramétrable) quand on importe les JDD de l'utilisateur qui se connecte (dans la table `gn_commons.cor_module_dataset`)
- Mise à jour des librairies Python Utils-Flask-SQLAlchemy (en version 0.1.1) et Utils-Flask-SQLAlchemy-Geo (en version 0.1.0) permettant de mettre en place les exports au format GeoPackage et corrigeant les exports de SHP contenant des géométries multiples

## Corrections

- Mise à jour des URL de la documentation utilisateur des modules, renvoyant vers <http://docs.geonature.fr>
- Validation : Correction de l'ouverture de la fiche d'information d'une observation (#858)
- Modification de l'attribution de la hauteur du composant `map-container` pour permettre d'adapter la hauteur de la carte si la hauteur d'un conteneur parent est modifié. Et que `<pnx-map height="100%">` fonctionne (#844 par @jbrieuclp)
- Mise à jour de la librairie python MarkupSafe en version 1.1, corrigeant un problème de setuptools (#881)
- Page Maintenance : Correction de l'affichage de l'image (par @jpm-cbna)
- Correction du multiselect du composant `pnx-nomenclatures` (#885 par @jpm-cbna)
- Correction de l'input ('coordinates') du composant `marker` (#901 par @jbrieuclp)
- Utilisation de NVM quand on installe les dépendances javascript (#926 par @jpm-cbna)

- Formulaire JDD : Correction de l’affichage de la liste des modules (#861)
- Correction de l’utilisation des paramètres du proxy (#944)

### Notes de version

Si vous mettez à jour GeoNature.

- Vous devez d’abord mettre à jour TaxHub en version 1.7.0
- Si vous mettez à jour TaxHub, vous pouvez mettre à jour Taxref en version 13. Il est aussi possible de le faire en différé, plus tard
- Vous pouvez mettre à jour UsersHub en version 2.1.2
- Exécuter le script SQL de mise à jour des nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update1.3.2to1.3.3.sql>).
- Si vous avez mis à jour Taxref en version 13, répercuter les évolutions au niveau des nomenclatures avec le script SQL [https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update\\_taxref\\_v13.sql](https://github.com/PnX-SI/Nomenclature-api-module/blob/master/data/update_taxref_v13.sql). Sinon vous devrez l’exécuter plus tard, après avoir mis à jour Taxref en version 13. Après avoir mis à jour Taxref en version 13, pensez à mettre à jour le paramètre `taxref_version` dans la table `gn_commons.t_parameters`.
- Exécuter le script SQL de mise à jour de la BDD de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.3.2to2.4.0.sql>)
- Installer les dépendances de la librairie Python WeasyPrint :

```
sudo apt-get install -y libcairo2
sudo apt-get install -y libpango-1.0-0
sudo apt-get install -y libpangocairo-1.0-0
sudo apt-get install -y libgdk-pixbuf2.0-0
sudo apt-get install -y libffi-dev
sudo apt-get install -y shared-mime-info
```

- Corriger l’utilisation des paramètres du proxy (#944) dans le fichier `backend/gunicorn_start.sh` en remplaçant les 2 lignes :

```
export HTTP_PROXY="$proxy_http"
export HTTPS_PROXY="$proxy_https"
```

par :

```
# Activation de la configuration des proxy si nécessaire
[[ -z "$proxy_http" ]] || export HTTP_PROXY="$proxy_http"
[[ -z "$proxy_https" ]] || export HTTPS_PROXY="$proxy_https"
```

- Vous pouvez supprimer les associations des observations de la synthèse aux zonages limitrophes, si vous n’avez pas d’observations sans géométrie (#719) :

```
DELETE FROM gn_synthese.cor_area_synthese cas
USING gn_synthese.synthese s, ref_geo.l_areas a
WHERE cas.id_synthese = s.id_synthese AND a.id_area = cas.id_area
AND public.ST_TOUCHES(s.the_geom_local,a.geom);
```

- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

## 10.24 2.3.2 (2020-02-24)

### Nouveautés

- Possibilité de charger les commandes d'un module dans les commandes de GeoNature
- Ajout de commentaires dans le fichier d'exemple de configuration `config/default_config.toml.example`

### Corrections

- Correction d'une incohérence dans le décompte des JDD sur la page d'accueil en leur appliquant le CRUVED (#752)
- Montée de version de la librairie `utils-flask-sqlalchemy-geo` pour compatibilité avec la version 1.0.0 du module d'export

## 10.25 2.3.1 (2020-02-18)

### Nouveautés

- Installation globale : Compatibilité Debian 10 (PostgreSQL 11, PostGIS 2.5)
- Installation globale : Passage à Taxhub 1.6.4 et UsersHub 2.1.1
- Utilisation généralisée des nouvelles librairies externalisées de sérialisation (<https://github.com/PnX-SI/Utils-Flask-SQLAlchemy> et <https://github.com/PnX-SI/Utils-Flask-SQLAlchemy-Geo>)
- Possibilité de régler le timeout de Gunicorn pour éviter le plantage lors de requêtes longues
- Ne pas zoomer sur les observations au premier chargement de la carte (#838)
- Leaflet-draw : Ajout de la possibilité de zoomer sur le point (par @joelclems)
- Ajout du nom vernaculaire dans les fiches d'information des relevés d'Occtax (par @FloVollmer / #826)

### Corrections

- Correction de l'installation de Node.js et npm par l'utilisation généralisée de nvm (#832 et #837)
- Fixation de la version de Node.js en 10.15.3 (dans le fichier `fronted/.nvmrc`)
- Ajout d'une référence de l'objet Leaflet `L` afin qu'il soit utilisé dans les modules et changement du typage de l'événement Leaflet `MouseEvent` en `L.LeafletMouseEvent`
- Fixation de la version de virtualenv en 20.0.1 (par @sogalgeeko)
- Corrections de typos dans la documentation d'administration (#840 - par @sogalgeeko)

### Notes de version

- Vous pouvez passer directement à cette version depuis la 2.2.x, mais en suivant les notes des versions intermédiaires (NB : il n'est pas nécessaire d'exécuter le script `migrate.sh` des versions précédentes)
- Installez pip3 et virtualenv:

```
sudo apt-get update
sudo apt-get install python3-pip
sudo pip3 install virtualenv==20.0.1
```

- Rajoutez la ligne `gun_timeout=30` au fichier `config/settings.ini` puis rechargez supervisor (`sudo supervisorctl reload`). Il s'agit du temps maximal (en seconde) autorisé pour chaque requête. A augmenter, si vous avez déjà rencontré des problèmes de timeout.



- Depuis le répertoire `frontend`, lancez la commande `nvm install`

## 10.26 2.3.0 - Occhab de Noël (2019-12-27)

### Nouveautés

- Développement du module Occhab (Occurrences d'habitats) basé sur une version minimale du standard SINP du même nom et s'appuyant sur le référentiel Habref du SINP (#735)
  - Consultation (carte-liste) des stations et affichage de leurs habitats
  - Recherche (et export) des stations par jeu de données, habitats ou dates
  - Saisie d'une station et de ses habitats
  - Possibilité de saisir plusieurs habitats par station
  - Saisie des habitats basée sur une liste pré-définie à partir d'Habref. Possibilité d'intégrer toutes les typologies d'habitat ou de faire des listes réduites d'habitats
  - Possibilité de charger un fichier GeoJson, KML ou GPX sur la carte et d'utiliser un de ses objets comme géométrie de station
  - Mise en place d'une API Occhab (Get, Post, Delete, Export stations et habitats et récupérer les valeurs par défaut des nomenclatures)
  - Calcul automatique des altitudes (min/max) et de la surface d'une station
  - Gestion des droits (en fonction du CRUVED de l'utilisateur connecté)
  - Définition des valeurs par défaut dans la BDD (paramétrable par organisme)
  - Possibilité de masquer des champs du formulaire
- Création d'un sous-module autonome ou intégré pour gérer l'API d'Habref (<https://github.com/PnX-SI/Habref-api-module>) pour :
  - Rechercher un habitat dans Habref (avec usage du trigramme pour la pertinence du résultat)
  - Obtenir les infos d'un habitat et de ses correspondances à partir de son `cd_hab`
  - Obtenir les habitats d'une liste (avec ou sans leur code en plus de leur nom et filtrable par typologie)
  - Obtenir la liste des typologies (filtrable par liste d'habitats)
- Mise à jour du module des nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module>) en version 1.3.2 incluant notamment :
  - Ajout de nomenclatures SINP concernant les habitats
  - Ajout d'une contrainte d'unicité sur la combinaison des champs `id_type` et `cd_nomenclature` de la table `t_nomenclatures`
- Association des JDD à des modules pour filtrer les JDD utilisés dans Occtax ou dans Occhab notamment (#399)
- Mise à jour de Angular 4 à Angular 7 (performances, ...) par @jbrieuclp
- Ajout d'une documentation utilisateur pour le module Synthèse : <http://docs.geonature.fr/user-manual.html#synthese> (par @amandine-sahl)
- OCCTAX : Amélioration importante des performances de la liste des relevés (par @jbrieuclp) (#690, #740)
- Améliorations des performances des exports de Occtax et de Synthèse et ajout d'index dans Occtax (par @gildeluermoz) (#560)

- Partage de scripts de sauvegarde de l'application et de la BDD dans `data/scripts/backup/` (par @gildeluermoz)
- Externalisation des librairies d'outils Flask et SQLAlchemy (<https://github.com/PnX-SI/Utils-Flask-SQLAlchemy> et <https://github.com/PnX-SI/Utils-Flask-SQLAlchemy-Geo>) pour pouvoir les factoriser et les utiliser dans d'autres applications. Cela améliore aussi les performances des jointures.
- SYNTHESE : Ajout d'un export de la liste des espèces (#805)
- SYNTHESE : Baser la portée de tous les exports (y compris Statuts) sur l'action E (#804)
- METADONNEES : Affichage des ID des JDD et CA
- OCCTAX : Conserver le fichier GPX ou GeoJSON chargé sur la carte quand on enchaîne des relevés et ajouter de la transparence sur les géométries utilisés dans les relevés précédents (#813)
- OCCTAX : Clarification de l'ergonomie pour ajouter un dénombrement sur un taxon (#780)
- Ajout des dates de création et de modification dans les tables `gn_monitoring.t_base_sites` et `gn_monitoring.t_base_visits` et triggers pour les calculer automatiquement
- Ajout des champs `geom_local`, `altitude_min` et `altitude_max` dans la table `gn_monitoring.t_base_sites` et triggers pour les calculer automatiquement (#812)
- Ajout des champs `id_dataset`, `id_module`, `id_nomenclature_obs_technique` et `id_nomenclature_grp_typ` dans la table `gn_monitoring.t_base_visits` (#812)
- Le composant générique `FileLayer` expose un `output` pour récupérer la géométrie sélectionnée (un observable de `MapService` était utilisé auparavant)
- Support des `markers` sur le composant `leaflet-draw`
- Possibilité de ne pas activer le composant `marker` au lancement lorsque celui-ci est utilisé (`input defaultEnable`)
- Ajout d'inputs `time`, `number`, `medias` et `datalist` au composant `DynamicForm` permettant de générer des formulaires dynamiques.
- Améliorations diverses du composant `DynamicForm` (par @joelclems)
- Ajout d'un paramètre dans le cas où le serveur se trouve derrière un proxy (`proxy_http` ou dans `proxy_https` dans `config/settings.ini`)
- Ajout d'une route permettant de récupérer la liste des rôles d'une liste à partir de son code (par @joelclems)

### Corrections

- MENU Side nav : Correction pour ne pas afficher les modules pour lesquels le paramètre `active_frontend` est `False` (#822)
- OCCTAX : Gestion de l'édition des occurrences où le JDD a été désactivé, en ne permettant pas de modifier le JDD (#694)
- OCCTAX : Correction d'une faiblesse lors de la récupération des informations taxonomiques d'un relevé (utilisation d'une jointure plutôt que l'API TaxHub) (#751)
- OCCTAX : Correction des longues listes de taxons dans les tooltip des relevés en y ajoutant un scroll (par @jbrieuclp) (#666)
- OCCTAX : Masquer le bouton `Télécharger` si l'utilisateur n'a pas de droits d'export dans le module (`E = 0`)
- OCCTAX : Correction de l'affichage des relevés dans la liste (#777)
- OCCTAX : Correction des exports quand on filtre sur un observateur en texte

- **SYNTHESE** : Filtre sur `date_max` en prenant `date_max <= 23:59:59` pour prendre en compte les observations avec un horaire (#778)
- **SYNTHESE** : Correction des boutons radios pour les filtres taxonomiques avancés basés sur les attributs TaxHub (#763)
- **SYNTHESE** : Correction de la recherche par `cd_nom` dans le composant `SearchTaxon` (#824)
- **VALIDATION** : Corrections mineures (par @jbrieuclp) (#715)
- **INSCRIPTION** : Correction si aucun champ additionnel n'a été ajouté au formulaire (par @jbrieuclp) (#746)
- **INSCRIPTION** : Correction de l'usage des paramètres `ENABLE_SIGN_UP` et `ENABLE_USER_MANAGEMENT` (#791)
- Simplification de l'écriture des logs dans le script `install_db.sh`
- Correction de l'installation des `requirements.txt` lors de l'installation d'un module (#764 par @joelclems)
- **COMMONS** : Modification des champs de `t_modules` de type `CHARACTER(n)` en `CHARACTER VARYING(n)` (`module_path`, `module_target`, `module_external_url`) (#799)
- **COMMONS** : Ajout de contraintes d'unicité pour les champs `module_path` et `module_code` de `t_modules`
- `pnx-geojson` : Amélioration du zoom, gestion des styles
- Migration des données GeoNature V1 vers V2 (`data/migrations/v1tov2/`) : ajustements mineurs

#### Notes de version

NB: La version 2.3.0 n'est pas compatible avec le module Dashboard. Si vous avez le module Dashboard installé, ne passez pas à cette nouvelle version. Compatibilité dans la 2.3.1.

- Lancer le script de migration qui va installer et remplir le nouveau schéma `ref_habitats` avec Habref et mettre à jour le schéma `ref_nomenclatures` :

```
cd /home/~whoami~/geonature/install/migration
chmod +x 2.2.1to2.3.0.sh
./2.2.1to2.3.0.sh
```

Vérifier que la migration s'est bien déroulée dans le fichier `var/log/2.2.1to2.3.0.log`.

- Lancer le script SQL de mise à jour de la BDD de GeoNature <https://raw.githubusercontent.com/PnX-SI/GeoNature/2.3.0/data/migrations/2.2.1to2.3.0.sql>
- Vous pouvez installer le nouveau module Occhab (Occurrences d'habitats) si vous le souhaitez :

```
cd /home/~whoami~/geonature/backend
source venv/bin/activate
geonature install_gn_module /home/~whoami~/geonature/contrib/gn_module_occhab /occhab
deactivate
```

- Lors de la migration (`/data/migrations/2.2.1to2.3.0.sql`), tous les JDD actifs sont associés par défaut au module Occtax (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.2.1to2.3.0.sql#L17-L22>). A chacun d'adapter si besoin, en retirant certains. Pour utiliser le module Occhab, vous devez y associer au moins un JDD.

## 10.27 2.2.1 (2019-10-09)

### Corrections

- La route de changement de mot de passe était désactivée par le mauvais paramètre (ENABLE\_SIGN\_UP au lieu de ENABLE\_USER\_MANAGEMENT)
- Désactivation du mode “enchaînement des relevés” en mode édition (#669). Correction effacement du même relevé (#744)
- Correction d’affichage du module métadonnées lorsque les AF n’ont pas de JDD pour des raisons de droit (#743)
- Diverses corrections de doublons d’import et de logs de débbugs (#742)
- Montée de version du sous-module d’authentification: 1.4.2

## 10.28 2.2.0 - Module utilisateurs (2019-09-18)

### Nouveautés

- Ajout d’interfaces et de paramètres de création de compte, de récupération de son mot de passe et d’administration de son profil, basé sur l’API UsersHub 2.1.0 (par @jbrieuclp et @TheoLechemia) #615
- Ajout d’une fonctionnalité de création automatique d’un CA et d’un JDD personnel lors de la validation d’un compte créé automatiquement (paramétrable)
- Amélioration du composant de création dynamique de formulaire (support de text-area, checkbox simple et multiple et exemple d’utilisation à partir de la conf GeoNature)
- Le composant ‘observateur’ permet de rechercher sur le nom ou le prénom (utilisation des RegEx) #567
- Mise à jour de Flask en version 1.1.1
- Nouvelle version du sous-module d’authentification (1.4.1), compatible avec UsersHub 2.1.0
- Mise à jour du sous-module de nomenclatures (version 1.3.0)
- Mise à jour et clarification du MCD (<http://docs.geonature.fr/admin-manual.html#base-de-donnees>) par @jpm-cbna
- Ajout d’une tutoriel vidéo d’installation dans la documentation (<https://www.youtube.com/watch?v=JYgH7cV9AjE>, par @olivier8064)

### Corrections

- Correction d’un bug sur les export CSV en utilisant la librairie Python standard `csv` (#733)
- SYNTHESE API : Passage de la route principale de récupération des données en POST plutôt qu’en GET (#704)
- SYNTHESE BDD : Suppression automatique des aires intersectées (`synthese.cor_area_synthese`) lorsqu’une observation est supprimée (DELETE CASCADE)
- SYNTHESE : Prise en compte du paramètre `EXPORT_ID_SYNTHESE_COL` (#707)
- OCCTAX : Correction d’une autocomplétion automatique erronée de la date max en mode édition (#706)
- VALIDATION : Améliorations des performances, par @jbrieuclp (#710)
- Prise en compte des sous-taxons pour le calcul des règles de sensibilité
- Correction des contraintes CHECK sur les tables liées à la sensibilité
- Complément et correction des scripts de migration `data/migrations/v1tov2`

- Correction et clarification de la documentation d'administration des listes de taxons et de sauvegarde et restauration de la BDD (par @lpofredc)
- Correction de la rotation des logs

#### Notes de version

- Passer le script de migration suivant: <https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.1.2to2.2.0.sql>
- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Si vous souhaitez activer les fonctionnalités de création de compte, veuillez lire **attentivement** cette documentation : <http://docs.geonature.fr/admin-manual.html#configuration-de-la-creation-de-compte>
- Si vous activez la création de compte, UsersHub 2.1.0 doit être installé. Voir sa [note de version](#).

## 10.29 2.1.2 (2019-07-25)

#### Corrections

- SYNTHESE : Correction d'une URL en dur pour la recherche de rangs taxonomiques
- OCC TAX : Affichage uniquement des JDD actifs
- VALIDATION : Abaissement de la limite d'affichage de données sur la carte par défaut + message indicatif
- Migration : Suppression d'un lien symbolique qui créait des liens en cascade
- Amélioration de la documentation (@dthonon)
- Amélioration de la rapidité d'installation du MNT grâce à la suppression d'un paramètre inutile
- BACKOFFICE : Correction d'une URL incorrecte et customisation

#### Notes de version

Ceci est une version corrective mineure. Si vous migrez depuis la 2.1.0, passez directement à cette version en suivant les notes de version de la 2.1.1.

## 10.30 2.1.1 (2019-07-18)

#### Nouveautés

- SYNTHESE: Factorisation du formulaire de recherche (utilisé dans le module synthese et validation)
- SYNTHESE: Simplification et correction du module de recherche avancée d'un taxon en le limitant à l'ordre (performances)
- SYNTHESE: Ajout d'un composant de recherche taxonomique avancé basé sur les rangs taxonomiques (modules synthese et validation), basé sur la nouvelle fonction `taxonomie.find_all_taxons_children` ajoutée à TaxHub
- Création d'un backoffice d'administration dans le coeur de GeoNature. Basé sur Flask-admin, les modules peuvent alimenter dynamiquement le backoffice avec leur configuration
- Mise en place d'une documentation développeur automatique de l'API à partir des docstring et des composants frontend, générée par Travis et désormais accessible à l'adresse <http://docs.geonature.fr> (#673)
- Amélioration de la documentation (triggers, installation, module validation)

- Suppression du module d'exemple, remplacé par un template de module ([https://github.com/PnX-SI/gn\\_module\\_template](https://github.com/PnX-SI/gn_module_template))
- Ajout d'un champ `validable` sur la table `gn_meta.t_datasets` controlant les données présentes dans le module VALIDATION ([https://github.com/PnX-SI/gn\\_module\\_validation/issues/31](https://github.com/PnX-SI/gn_module_validation/issues/31))
- VALIDATION: Lister toutes les données de la synthèse ayant un `uuid_sinp` dans le module validation, et plus seulement celles qui ont un enregistrement dans `gn_commons.t_validations`
- VALIDATION: On ne liste plus les `id_nomenclatures` des types de validation à utiliser, dans la configuration du module. Mais on utilise toutes les nomenclatures activées du type de nomenclature `STATUT_VALID`. ([https://github.com/PnX-SI/gn\\_module\\_validation/issues/30](https://github.com/PnX-SI/gn_module_validation/issues/30))
- Ajout de tests sur les ajouts de JDD et CA
- Ajout d'une fonctionnalité d'envoi d'email via Flask-Mail dans le coeur de GeoNature
- Amélioration des performances: ajout d'index sur Occtax et Metadonnées
- Script d'import des métadonnées à partir du webservice MTD de l'INPN (@DonovanMaillard)
- Complément, correction et compatibilité 2.1.0 des scripts de migration `data/migrations/v1tov2`

### Corrections

- Nombreuses corrections du module de validation (non utilisation des `id_nomenclature`, simplification des vues et de la table `gn_commons.t_validations`)
- Ordonnancement de listes déroulantes (#685)
- OCCTAX : correction de l'édition d'un relevé à la géométrie de type Polyline (#684)
- OCCTAX : correction l'édition et du contrôle conditionnel des champs de "preuves" (preuve d'existence numérique / non numérique) (#679)
- OCCTAX : correction du parametre `DATE_FORM_WITH_TODAY` non pris en compte (#670)
- OCCTAX: correction de la `date_max` non remplie lorsque `DATE_FORM_WITH_TODAY = false`
- OCCTAX: correction d'un bug lors de l'enchaînement de relevé lorsque l'heure est remplie
- SYNTHESE: correction des doublons lorsqu'il y a plusieurs observateurs
- Correction du composant `dynamicForm` sur les champs de recherche de type texte (recherche sur Preuve numérique) (#530)
- Désactivation du mode "enchaîner les relevés" en mode édition (#699)
- Correction de `gn_monitoring` : utiliser `gn_commons.t_modules` à la place de `utilisateurs.t_applications` pour associer des sites de suivi à des modules
- Fix de SQLAlchemy 1.3.3 et jointure sur objet Table
- Le trigger remplissant `cor_area_synthese` en intersectant `gn_synthese.synthese` avec `ref_geo.l_areas` ne prend plus que les zonages ayant le champs `enabled=true`
- Correction `dict()` et version de Python (par @jpm-cbna)
- MAJ de sécurité de Bootstrap (en version 4.3.1)
- L'ancien module export du coeur est enlevé en vue de la sortie du nouveau module export

### Notes de version

- Passer TaxHub en version 1.6.3 (<https://github.com/PnX-SI/TaxHub/releases/tag/1.6.3>)
- Passer le script de migration `data/2.1.0to2.1.1.sql`

- Si vous aviez modifier les `id_nomenclature` dans la surcouche de la configuration du module validation, supprimer les car on se base maintenant sur les `cd_nomenclature`
- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Nouvelle localisation de la doc : <http://docs.geonature.fr>

## 10.31 2.1.0 - Module validation (2019-06-01)

### Nouveautés

- Intégration du module Validation dans GeoNature (développé par @JulienCorny, financé par @sig-pnrnm)
- Ajout de tables, règles et fonctions permettant de calculer la sensibilité des occurrences de taxon de la synthèse (#284)
- Occtax - Possibilité d'enchaîner les saisies de relevés et de garder les informations du relevé (#633)
- Occtax - Amélioration de l'ergonomie de l'interface MapList pour clarifier la recherche et l'ajout d'un relevé + ajout compteur (#467)
- Révision de l'interface du module Métadonnées, listant les cadres d'acquisition et leurs jeux de données (par @jbrieuclp)
- Ajout d'un mécanisme du calcul des taxons observés par zonage géographique (#617)
- Les mailles INPN (1, 5, 10km) sont intégrées à l'installation (avec un paramètre)
- Statistiques de la page d'accueil - Ajout d'un paramètre permettant de les désactiver (#599)
- Occtax - Date par défaut paramétrable (#351)
- Support des géométries multiples (MultiPoint, MultiPolygone, MultiLigne) dans la synthèse et Occtax (#609)
- Synthèse - Affichage des zonages intersectés dans un onglet séparé (#579)

### Corrections

- Révision complète des scripts de migration de GeoNature v1 à v2 (`data/migrations/v1tov2`)
- Masquer l'export du module Synthèse si son CRUVED est défini à 0 (#608)
- Correction de la vérification du CRUVED du module METADONNEES (#601)
- Correction de la vérification du CRUVED lorsque `get_role = False`
- Correction de la traduction sur la page de connexion (par @jbrieuclp)
- Occtax - Retour du composant GPS permettant de charger un marker à partir de coordonnées X et Y (#624)
- Correction lors d'import de fichier GPX ayant une altitude (#631)
- Occtax - Correction du filtre Observateur texte libre (#598)
- Métadonnées - Inversion des domaines terrestre/marin (par @xavyeah39)
- Métadonnées - Correction de l'édition des cadres d'acquisition (#654, par @DonovanMaillard)
- Mise à jour de sécurité de Jinja2 et SQLAlchemy

### Notes de version

- Vous pouvez passer directement à cette version, mais en suivant les notes des versions intermédiaires

- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Lancer le script de migration de la base de données :

Cette nouvelle version de GeoNature intègre les mailles INPN (1, 5, 10km) dans le référentiel géographique. Si vous ne souhaitez pas les installer, lancer le script ci dessous en passant le paramètre `no-grid`

```
cd /home/~whoami~/geonature/data/migrations
# avec les mailles
./2.0.1to2.1.0.sh
# sans les mailles:
# ./2.0.1to2.1.0.sh no-grid
```

- Installer le module VALIDATION si vous le souhaitez :

Se placer dans le virtualenv de GeoNature

```
cd /home/~whoami~/geonature/backend
source venv/bin/activate
```

Lancer la commande d'installation du module puis sortir du virtualenv

```
geonature install_gn_module /home/~whoami~/geonature/contrib/gn_module_validation/
↪ /validation
deactivate
```

## 10.32 2.0.1 (2019-03-18)

### Nouveautés

- Développement : ajout d'une fonction de génération dynamique de requête SQL (avec vérification et cast des types)
- Synthèse : Ajout d'un message indiquant que le module affiche les dernières observations par défaut

### Corrections

- Synthèse : correction du filtre CRUVED pour les portées 1 et 2 sur la route `synthese/for_web` (#584)
- Synthèse : correction du bug lorsque la géométrie est null (#580)
- Synthèse : Correction de la redirection vers le module de saisie (#586)
- Synthèse : Correction de la valeur par défaut de la nomenclature `STATUT_OBS` (Présent au lieu de NSP)
- Configuration carto : correction du bug d'arrondissement des coordonnées géographiques (#582)
- Correction du trigger de calcul de la geom locale
- Recréation de la vue `pr_occtax.export_occtax_sinp` qui avait été supprimée lors de la migration RC3 vers RC4
- Correction de la vue `pr_occtax.v_releve_list`
- Correction ajout rang et `cd_nom` sur l'autocomplete de la synthèse, absent dans le script de migration
- DEPOBIO : Correction de la déconnexion au CAS INPN
- Occtax et Metadata: correction lors de la mise à jour d'un élément (Merge mal géré par SQLAlchemy lorsqu'on n'a pas une valeur NULL) (#588)



- Composant “jeu de données” : retour à l’affichage du nom long (#583)
- Amélioration du style du composant multiselect
- Metadata : formulaire cadre d’acquisition - listage uniquement des cadres d’acquisition parent pour ne pas avoir de cadres d’acquisition imbriqués
- Ajouts de tests automatisés complémentaires

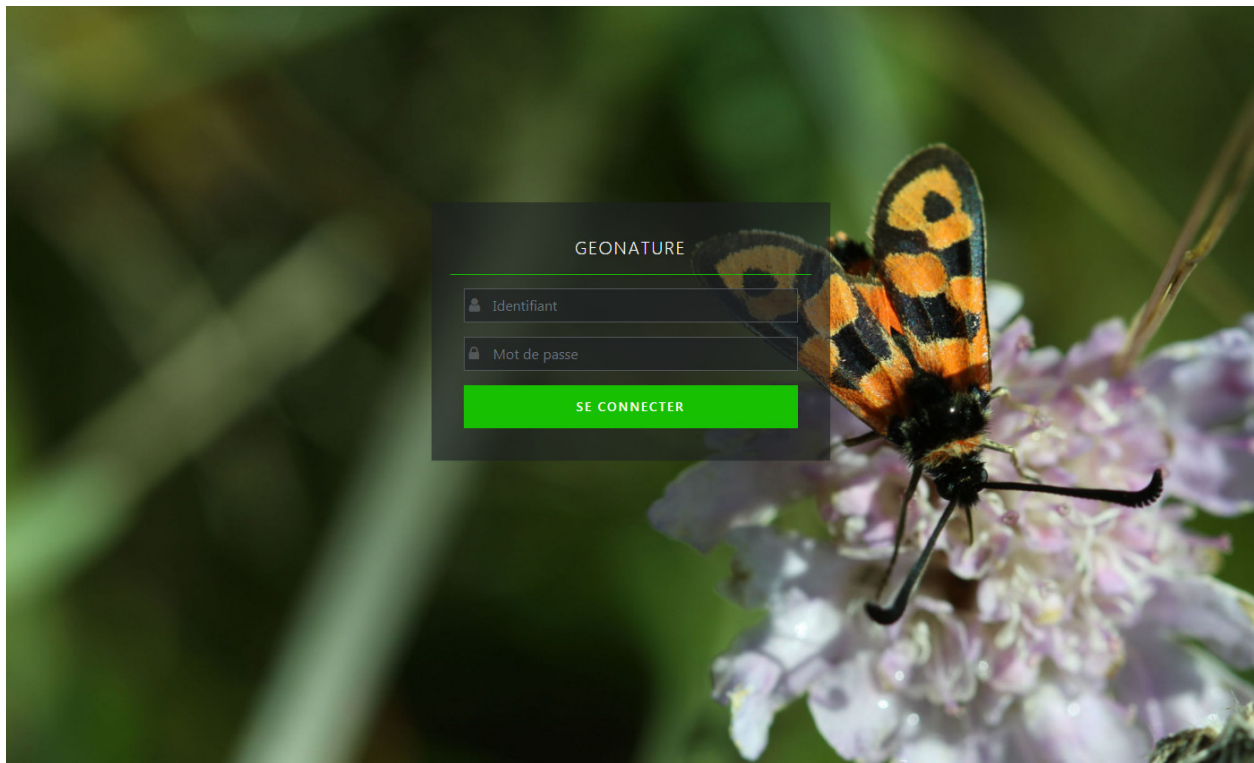
#### Notes de version

- Vous pouvez passer directement à cette version, mais en suivant les notes des versions intermédiaires
- Exécuter le script de migration SQL du sous-module Nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module/blob/1.2.4/data/update1.2.3to1.2.4.sql>)
- Exécuter le script de migration SQL de GeoNature (<https://github.com/PnX-SI/GeoNature/blob/master/data/migrations/2.0.0to2.0.1.sql>)
- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

## 10.33 2.0.0 - La refonte (2019-02-28)

La version 2 de GeoNature est une refonte complète de l’application.

- Refonte technologique en migrant de PHP/Symfony/ExtJS/Openlayers à Python3/Flask/Angular4/Leaflet
- Refonte de l’architecture du code pour rendre GeoNature plus générique et modulaire
- Refonte de la base de données pour la rendre plus standardisée, plus générique et modulaire
- Refonte ergonomique pour moderniser l’application



Pour plus de détails sur les évolutions apportées dans la version 2, consultez les détails des versions RC (Release Candidate) ci-dessous.

### Nouveautés

- Possibilité de charger un fichier (GPX, GeoJson ou KML) sur la carte pour la saisie dans le module Occtax (#256)
- Ajout d'un moteur de recherche de lieu (basé sur l'API OpenStreetMap Nominatim) sur les modules cartographiques (#476)
- Intégration du plugin leaflet markerCluster permettant d'afficher d'avantage d'observations sur les cartes et de gérer leurs superposition (#559)
- Synthèse : possibilité de grouper plusieurs types de zonages dans le composant `pnx-areas`
- Design de la page de login
- Intégration d'un bloc stat sur la page d'accueil
- Ajout d'un export des métadonnées dans la synthèse
- Centralisation de la configuration cartographique dans la configuration globale de GeoNature (`geonature_config.toml`)
- Cartographie : zoom sur l'emprise des résultats après une recherche
- Migration de la gestion des métadonnées dans un module à part : 'METADATA' (#550)
- Export vue synthèse customisable (voir doc)
- Lien vers doc par module (customisables dans `gn_commons.t_modules`) (#556)
- Ajout du code du département dans les filtres par commune (#555)
- Ajout du rang taxonomique et du `cd_nom` après les noms de taxons dans la recherche taxonomique (#549)
- Mise à jour des communes fournies lors de l'installation (IGN admin express 2019) (#537)
- Synthèse : Ajout du filtre par organisme (#531), affichage des acteurs dans les fiches détail et les exports
- Synthèse: possibilité de filtrer dans les listes déroulantes des jeux de données et cadres d'acquisition
- Filtre de la recherche taxonomique par règne et groupe INPN retiré des formulaires de recherche (#531)
- Suppression du champ validation dans le schéma de BDD Occtax car cette information est stockée dans la table verticale `gn_commons.t_validations` + affichage du statut de validation dans les fiches Occtax et Synthèse
- Ajout d'une vue `gn_commons.v_lastest_validation` pour faciliter la récupération du dernier statut de validation d'une observation
- Suppression de toutes les références à `taxonomie.bib_noms` en vue de le supprimer de TaxHub
- Séparation des commentaires sur l'observation et sur le contexte (relevé) dans la Synthèse et simplification des triggers de Occtax vers Synthèse (#478)
- Nouveau logo GeoNature (#346)

### Corrections

- Améliorations importantes des performances de la synthèse (#560)
- Synthèse : correction liée aux filtres multiples et aux filtres géographiques de type cercle
- Ajout d'une contrainte `DELETE CASCADE` entre `ref_geo.li_municipalities` et `ref_geo.l_areas` (#554)

- Occtax : possibilité de saisir un dénombrement égal à 0 (cas des occurrences d'absence)
- Occtax : retour à l'emprise cartographique précédente lorsqu'on enchaîne les relevés (#570)
- Occtax : correction de l'automplissage du champ `hour_max` lors de l'édition d'un relevé
- Divers compléments de la documentation (merci @jbdesbas, @xavyeah39 et @DonovanMaillard)
- Ajout de contraintes d'unicité sur les `UUID_SINP` pour empêcher les doublons (#536)
- Corrections et compléments des tests automatiques
- Amélioration de l'installation des modules GeoNature

## Notes de version

### 1. Pour les utilisateurs utilisant la version 1 de GeoNature :

Il ne s'agit pas de mettre à jour GeoNature mais d'en installer une nouvelle version. En effet, il s'agit d'une refonte complète.

- Sauvegarder toutes ses données car l'opération est complexe et non-automatisée
- Passer à la dernière version 1 de GeoNature (1.9.1)
- Passer aux dernières versions de UsersHub et TaxHub
- Installer GeoNature standalone ou refaire une installation complète
- Adaptez les scripts présents dans `/data/migrations/v1tov2` et exécutez-les pas à pas. Attention ces scripts ont été faits pour la version 2.0.0-rc.1 et sont donc à ajuster, tester, compléter et adapter à votre contexte

### 2. Pour les utilisateurs utilisant une version RC de GeoNature 2 :

Veuillez bien lire ces quelques consignes avant de vous lancer dans la migration.

- Vous pouvez passer directement à cette version, mais en suivant les notes des versions intermédiaires.
- Les personnes ayant configuré leur fichier `map.config.ts` devront le répercuter dans `geonature_config.toml`, suite à la centralisation de la configuration cartographique (voir [https://github.com/PnX-SI/GeoNature/blob/2.0.0/config/default\\_config.toml.example](https://github.com/PnX-SI/GeoNature/blob/2.0.0/config/default_config.toml.example) section [MAPCONFIG]).
- La configuration des exports du module synthèse a été modifiée (voir <http://docs.geonature.fr/user-manual.html#synthese>). Supprimer la variable ``[SYNTHESE.EXPORT\_COLUMNS]`` dans le fichier `geonature_config.toml`. Voir l'exemple dans le fichier (voir [https://github.com/PnX-SI/GeoNature/blob/2.0.0/config/default\\_config.toml.example](https://github.com/PnX-SI/GeoNature/blob/2.0.0/config/default_config.toml.example) section) pour configurer les exports.
- Supprimer la variable `COLUMNS_API_SYNTHESE_WEB_APP` si elle a été ajoutée dans le fichier `geonature_config.toml`.
- Pour simplifier son édition, le template personnalisable de la page d'accueil (`frontend/src/custom/components/introduction/introduction.component.html`) a été modifié (la carte des 100 dernière observations n'y figure plus). Veuillez supprimer tout ce qui se situe à partir de la ligne 21 (`<div class="row row-0">`) dans ce fichier.
- Exécuter le script de migration SQL: <https://github.com/PnX-SI/GeoNature/blob/2.0.0/data/migrations/2.0.0rc4.2to2.0.0.sql>.
- Le backoffice de gestion des métadonnées est dorénavant un module GeoNature à part. Le script migration précédemment lancé prévoit de mettre un CRUVED au groupe\_admin et groupe\_en\_poste pour le nouveau module METADATA. Les groupes nouvellement créés par les administrateurs et n'ayant de CRUVED pour l'objet METADATA (du module Admin), se retrouvent avec le CRUVED hérité de GeoNature. L'administrateur devra changer lui-même le CRUVED de ces groupes pour le nouveau module METADATA via le backoffice des permissions.

- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>).

## 10.34 2.0.0-rc.4.2 (2019-01-23)

### Nouveautés

- Mise en place de logs rotatifs pour éviter de surcharger le serveur
- Centralisation des logs applicatifs dans le dossier `var/log/gn_errors.log` de GeoNature

### Corrections

- Synthèse - Correction et amélioration de la gestion des dates (#540)
- Amélioration des tests automatisés
- Correction et complément ds scripts d'installation des modules GeoNature
- Remplacement de `gn_monitoring.cor_site_application` par `gn_monitoring.cor_site_module`
- Complément des documentations de customisation, d'administration et de développement
- Ajout d'une documentation de migration de données Serena vers GeoNature (<https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/serena>) par @xavyeah39

### Note de version

- Vous pouvez passer directement à cette version, mais en suivant les notes des versions intermédiaires
- Exécutez la mise à jour de la BDD GeoNature (`data/migrations/2.0.0rc4.1to2.0.0rc4.2.sql`)
- Depuis la version 2.0.0-rc.4, on ne stocke plus les modules de GeoNature dans `utilisateurs.t_applications`. On ne peut donc plus associer les sites de suivi de `gn_monitoring` à des applications, utilisé par les modules de suivi (Flore, habitat, chiro). Le mécanisme est remplacé par une association des sites de suivi aux modules. La création de la nouvelle table est automatisée (`data/migrations/2.0.0rc4.1to2.0.0rc4.2.sql`), mais pas la migration des éventuelles données existantes de `gn_monitoring.cor_site_application` vers `gn_monitoring.cor_site_module`, à faire manuellement.
- Afin que les logs de l'application soient tous écrits au même endroit, modifier le fichier `geonature-service.conf` (`sudo nano /etc/supervisor/conf.d/geonature-service.conf`). A la ligne `stdout_logfile`, remplacer la ligne existante par `stdout_logfile = /home/<MON_USER>/geonature/var/log/gn_errors.log` (en remplaçant `<MON_USER>` par votre utilisateur linux).
- Vous pouvez également mettre en place un système de logs rotatifs (système permettant d'archiver les fichiers de logs afin qu'ils ne surchargent pas le serveur - conseillé si votre serveur a une capacité disque limitée). Créer le fichier suivant `sudo nano /etc/logrotate.d/geonature` puis copiez les lignes suivantes dans le fichier nouvellement créé (en remplaçant `<MON_USER>` par votre utilisateur linux)

```
/home/<MON_USER>/geonature/var/log/*.log {
daily
rotate 8
size 100M
create
compress
}
```

Exécutez ensuite la commande `sudo logrotate -f /etc/logrotate.conf`

- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

## 10.35 2.0.0-rc.4.1 (2019-01-21)

### Corrections

- Mise à jour des paquets du frontend (#538)
- Correction d'un conflit entre Marker et Leaflet-draw
- Utilisation du paramètre `ID_APP` au niveau de l'application
- Corrections mineures diverses

### Note de version

- Sortie de versions correctives de UsersHub (2.0.2 - <https://github.com/PnX-SI/UsersHub/releases>) et TaxHub (1.6.1 - <https://github.com/PnX-SI/TaxHub/releases>) à appliquer aussi
- Vous pouvez vous référer à la documentation globale de mise à jour de GeoNature RC3 vers RC4 par @DonovanMaillard (<https://github.com/PnX-SI/GeoNature/blob/master/docs/update-all-RC3-to-RC4.rst>)

## 10.36 2.0.0-rc.4 (2019-01-15)

### Nouveautés

- Intégration de la gestion des permissions (CRUVED) dans la BDD de GeoNature, géré via une interface d'administration dédié (#517)
- Mise en place d'un système de permissions plus fin par module et par objet (#517)
- Mise en place d'un mécanisme générique pour la gestion des permissions via des filtres : filtre de type portée (SCOPE), taxonomique, géographique etc... (#517)
- Compatibilité avec UsersHub version 2
- L'administration des permissions ne propose que les rôles qui sont actif et qui ont un profil dans GeoNature
- Ajout du composant Leaflet.FileLayer dans le module Synthèse pour pouvoir charger un GeoJSON, un GPS ou KML sur la carte comme géométrie de recherche (#256)
- Ajout et utilisation de l'extension PostgreSQL `pg_trgm` permettant d'améliorer l'API d'autocomplétion de taxon dans la synthèse, en utilisant l'algorithme des trigrammes (<http://si.ecrins-parcnational.com/blog/2019-01-fuzzy-search-taxons.html>), fonctionnel aussi dans les autres modules si vous mettez à jour TaxHub en version 1.6.0.
- Nouvel exemple d'import de données historiques vers GeoNature V2 : <https://github.com/PnX-SI/Ressources-techniques/blob/master/GeoNature/V2/2018-12-csv-vers-synthese-FLAVIA.sql> (par @DonovanMaillard)
- Complément de la documentation HTTPS et ajout d'une documentation Apache (par @DonovanMaillard, @RomainBaghi et @lpofredc)

### Corrections

- Correction de l'id\_digitiser lors de la mise à jour (#481)
- Corrections multiples de la prise en compte du CRUVED (#496)
- Deconnexion apres inactivité de l'utilisateur (#490)

- Suppression des heures au niveau des dates de l'export occtax (#485)
- Correction du message d'erreur quand on n'a pas de JDD (#479)
- Correction du champs commentaire dans les exports d'Occtax séparé entre relevé et occurrence (#478)
- Correction des paramètres de la fonction `GenericQuery.build_query_filter()` (par @patkap)
- Correction de l'administration des métadonnées (#466 #420)
- Métadonnées (JDD et CA) : ne pas afficher les utilisateurs qui sont des groupes dans les acteurs
- Ajout d'un champs dans la Synthèse permettant de stocker de quel module provient une occurrence et fonctions SQL associées (#412)
- Amélioration du style des champs obligatoires
- Améliorations mineures de l'ergonomie d'Occtax
- Correction du spinner qui tournait en boucle lors de l'export CSV de la Synthèse (#451)
- Correction des tests automatisés
- Amélioration des performances des intersections avec les zonages de `ref_geo.l_areas`
- Complément de la documentation de développement
- Simplification de la configuration des gn\_modules
- Occtax : ordonnancement des observation par date (#467)
- Occtax : Remplissage automatique de l'heure\_max à partir de l'heure\_min (#522)
- Suppression des warnings lors du build du frontend
- Correction de l'installation des modules GeoNature
- Ajout d'un message quand on n'a pas accès à une donnée d'un module
- Affichage du nom du module dans le Header (#398)
- Correction des outils cartographiques dans Occtax
- Correction complémentaire des styles des lignes sans remplissage (#458)
- MaplistService : correction du zoom sur les polygones et polygones
- Composant Areas et Municipalities : remise à zéro de la liste déroulante quand on efface la recherche ou remet à jour les filtres
- Composant Taxonomy : la recherche autocomplétée est lancée même si on tape plus de 20 caractères. Le nombre de résultat renvoyé est désormais paramétrable (#518)
- Limitation du nombre de connexions à la BDD en partageant l'instance DB avec les sous-modules
- Installation : utilisation d'un répertoire `tmp` local et non plus au niveau système pour limiter les problèmes de droits (#503)
- Evolution du template d'exemple de module GeoNature ([https://github.com/PnX-SI/GeoNature/tree/master/contrib/module\\_example](https://github.com/PnX-SI/GeoNature/tree/master/contrib/module_example)) pour utiliser l'instance DB et utiliser les nouveaux décorateurs de permissions (CRUVED)

#### Note de version

- Si vous effectuez une migration de GeoNature RC3 vers cette nouvelle version, il est nécessaire d'avoir installé UsersHub version 2.x au préalable. Suivez donc sa documentation (<https://github.com/PnX-SI/UsersHub/releases>) avant de procéder à la montée de version de GeoNature.

- Exécuter la commande suivante pour ajouter l'extension `pg_trgm`, en remplaçant la variable `$db_name` par le nom de votre BDD : `sudo -n -u postgres -s psql -d $db_name -c "CREATE EXTENSION IF NOT EXISTS pg_trgm;"`
- Mettez à jour TaxHub en version 1.6.0 pour bénéficier de l'amélioration de la recherche taxonomique dans tous les modules
- Exécutez la mise à jour de la BDD GeoNature (`data/migrations/2.0.0rc3.1-to-2.0.0rc4.sql`)
- Suivez ensuite la procédure classique de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

#### Note développeurs

- Vous pouvez faire évoluer les modules GeoNature en utilisant l'instance DB de GeoNature pour lancer les scripts d'installation (#498)
- Il n'est plus nécessaire de définir un `id_application` dans la configuration des modules GeoNature.
- La gestion des permissions a été revue et est désormais internalisée dans GeoNature (voir <http://docs.geonature.fr/development.html#developpement-backend>), il est donc nécessaire d'utiliser les nouveaux décorateurs décrit dans la doc pour récupérer le CRUVED.

## 10.37 2.0.0-rc.3.1 (2018-10-21)

#### Corrections

- Correction du script `install_all.sh` au niveau de la génération de la configuration Apache de TaxHub et UsersHub (#493)
- Suppression du `Servername` dans la configuration Apache de TaxHub du script `install_all.sh`
- Complément de la documentation de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)

#### Notes de version

- Si vous migrez depuis une version 2.0.0-rc.2, installez directement cette version corrective plutôt que la 2.0.0-rc.3, mais en suivant les notes de versions de la 2.0.0-rc.3
- Pour mettre en place la redirection de TaxHub sans `/`, consultez sa documentation <https://taxhub.readthedocs.io/fr/latest/installation.html#configuration-apache>
- Le script `install_all.sh` actuel ne semble pas fonctionner sur Debian 8, problème de version de PostGIS qui ne s'installe pas correctement

## 10.38 2.0.0-rc.3 (2018-10-18)

- Possibilité d'utiliser le MNT en raster ou en vecteur dans la BDD (+ doc MNT) #439 (merci @mathieubossaert)
- `INSTALL_ALL` - gestion du format date du serveur PostgreSQL (#435)
- `INSTALL_ALL` - Amélioration de la conf Apache de TaxHub pour gérer son URL sans `/` à la fin
- Dessin cartographique d'une autre couleur (rouge) que les observations (bleu)
- Occtax : retour au zoom précédent lors de l'enchaînement de relevé (#436)
- Occtax : observateur rempli par défaut avec l'utilisateur connecté (#438)
- Prise en compte des géométries nulles dans la fonction `serializegeofn`



- Gestion plus complète des données exemple intégrées ou non lors de l'installation (#446)
- Complément des différentes documentations
- Complément FAQ (#441)
- Documentation de la customisation (merci @DonovanMaillard)
- Amélioration de l'architecture du gn\_module d'exemple
- Clarification de la configuration des gn\_modules
- Lire le fichier `VERSION` pour l'afficher dans l'interface (#421)
- Utilisation de la vue `export_occtax_sinp` et non plus `export_occtax_dlb` par défaut pour les exports Occtax (#462)
- Complément et correction des vues `export_occtax_sinp` et `export_occtax_dlb` (#462)
- Mise à jour de Marshmallow (2.5.0 => 2.5.1)
- Améliorations des routes de `gn_monitoring` et de la configuration des modules de suivi pour pouvoir utiliser le nom d'une application plutôt que son identifiant
- Export Synthèse - Remplacement de la barre de téléchargement par un spinner (#451)

### Corrections

- Doc Import niveau 2 : Corrections et compléments
- Correction du trigger Occtax > Synthèse qui met à jour le champs `gn_synthese.observers_txt` et les commentaires (#448 et #459)
- Correction et amélioration de la fonction `install_gn_module`
- Correction coquille dans le modèle `gn_monitoring` et la fonction `serializegeofn`
- Installation uniquement sur un environnement 64 bits (documentation + vérification) #442 (merci @jbrieuclp et @sig-pnrm)
- Correction et découpage des scripts de mise à jour de la BDD depuis la version Beta5
- Correction de l'édition des `date_debut` et `date_fin` de Occtax (#457)
- Correction des exports depuis la Synthèse et intégration de la géométrie des observations (#461 et #456)
- Ne pas remplir `pr_occtax.cor_role_relevés_occtax` si `observers_txt = true` (#463)
- Edition d'un relevé Occtax - Ne pas recalculer l'altitude existante (#424)
- Correction de l'activation du formulaire Occtax après localisation du relevé (#469 et #471)
- Carte - Enlever le remplissage des lignes (#458)
- Amélioration du script de mise à jour de GeoNature (`install/migration/migration.sh`) (#465)
- Suppression d'un doublon dans le modèle de `gn_commons.t_modules` (merci @lpofredc)

### Autres

- Mise à jour de TaxHub (Doc utilisateur, configuration Apache, script d'import des médias depuis API INPN Taxref et remise à zéro des séquences)
- Script de migration des données SICEN (ObsOcc) vers GeoNature : <https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/sicen>
- Script d'import continu depuis une BDD externe vivante (avec exemple SICEN) : <https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration/generic>



- Module Suivi Flore Territoire fonctionnel et installable ([https://github.com/PnX-SI/gn\\_module\\_suivi\\_flore\\_territoire](https://github.com/PnX-SI/gn_module_suivi_flore_territoire))
- Module Suivi Chiro fonctionnel et installable ([https://github.com/PnCevennes/gn\\_module\\_suivi\\_chiro](https://github.com/PnCevennes/gn_module_suivi_chiro)) ainsi que son Frontend générique pour les protocoles de suivi ([https://github.com/PnCevennes/projet\\_suivis\\_frontend/](https://github.com/PnCevennes/projet_suivis_frontend/))
- Ebauche d'un module pour les protocoles CMR (Capture-Marquage-Recapture) : [https://github.com/PnX-SI/gn\\_module\\_cmr](https://github.com/PnX-SI/gn_module_cmr)
- MCD du module Suivi Habitat Territoire ([https://github.com/PnX-SI/gn\\_module\\_suivi\\_habitat\\_territoire](https://github.com/PnX-SI/gn_module_suivi_habitat_territoire))
- MCD du module Flore Prioritaire ([https://github.com/PnX-SI/gn\\_module\\_flore\\_prioritaire](https://github.com/PnX-SI/gn_module_flore_prioritaire))
- Consolidation du backend et premiers développements du frontend de GeoNature-citizen (<https://github.com/PnX-SI/GeoNature-citizen>)
- Création d'un script expérimental d'installation de GeoNature-atlas compatible avec GeoNature V2 dt pouvant utiliser son schéma `ref_geo` pour les communes, le territoire et les mailles ([https://github.com/PnX-SI/GeoNature-atlas/blob/develop/install\\_db\\_gn2.sh](https://github.com/PnX-SI/GeoNature-atlas/blob/develop/install_db_gn2.sh))

#### Notes de version

- Suivez la procédure standard de mise à jour de GeoNature (<http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>)
- Exécutez l'update de la BDD GeoNature (`data/migrations/2.0.0rc2-to-2.0.0rc3.sql`)
- Il est aussi conseillé de mettre à jour TaxHub en 1.5.1 (<https://github.com/PnX-SI/TaxHub/releases>) ainsi que sa configuration pour qu'il fonctionne sans / à la fin de son URL
- Attention, si vous installez cette version avec le script global `install_all.sh`, il créé un problème dans la configuration Apache de UserHub (`/etc/apache2/sites-available/usershub.conf`) et supprime tous les /. Les ajouter sur la page de la documentation de UsersHub (<https://github.com/PnX-SI/UsersHub/blob/master/docs/installation.rst#configuration-apache>) puis relancer Apache (<https://github.com/PnX-SI/GeoNature-atlas/blob/develop/docs/installation.rst>). Il est conseillé d'installer plutôt la version corrective.

## 10.39 2.0.0-rc.2 (2018-09-24)

#### Nouveautés

- Script `install_all.sh` compatible Ubuntu (16 et 18)
- Amélioration du composant Download
- Amélioration du ShapeService
- Compléments de la documentation
- Intégration de la documentation Développement backend dans la documentation
- Nettoyage du code
- Mise à jour de la doc de l'API : <https://documenter.getpostman.com/view/2640883/RWapSkTw>
- Configuration de la carte (`frontend/src/conf/map.config.ts`) : OSM par défaut car OpenTopoMap ne s'affiche pas à petite échelle

#### Corrections

- Correction du script `install/migration/migration.sh`

- Ne pas afficher le debug dans le recherche de la synthèse
- Correction du bug de déconnexion entre TaxHub et GeoNature (#423)
- Correction de la fiche info d'Occtax
- Champs Multiselect : Ne pas afficher les valeurs sélectionnées dans la liste quand on modifie un objet
- Trigger Occtax vers Synthèse : Correction des problèmes d'heure de relevés mal copiés dans la Synthèse
- Correction des altitudes (non abouti) (#424)
- Données exemple : Suppression de l'observers\_txt dans la synthèse
- Suppression d'un id\_municipality en dur dans une route
- Suppression de la librairie Certifi non utilisée

#### Notes de version

- Suivez la procédure standard de mise à jour de GeoNature
- Exécuter l'update de la BDD GeoNature (data/migrations/2.0.0rc1-to-2.0.0rc2.sql)

## 10.40 2.0.0-rc.1 (2018-09-21)

La version 2 de GeoNature est une refonte complète de l'application.

- Refonte technologique en migrant de PHP/Symfony/ExtJS/Openlayers à Python3/Flask/Angular4/Leaflet
- Refonte de l'architecture du code pour rendre GeoNature plus générique et modulaire
- Refonte de la base de données pour la rendre plus standard, plus générique et modulaire
- Refonte ergonomique pour moderniser l'application

Présentation et suivi des développements : <https://github.com/PnX-SI/GeoNature/issues/168>

#### Accueil

- Message d'introduction customisable
- Carte des 100 dernières observations
- CSS général de l'application surcouchable

#### Occtax

Module permettant de saisir, consulter, rechercher et exporter des données Faune, Flore et Fonge de type Contact selon le standard Occurrences de taxon du SINP (<https://inpn.mnhn.fr/telechargement/standard-occurrence-taxon>).

- Développement des formulaires de saisie, page de recherche, fiche détail, API, CRUVED et export
- Possibilité de masquer ou afficher les différents champs dans le formulaire Occtax (#344)
- Développement du formulaire de manière générique pour pouvoir réutiliser ses différents éléments dans d'autres modules sous forme de composants Angular
- Configuration possible du module (Niveau de zoom, champs affichées, export...)
- Ajout des nomenclatures dans les filtres d'Occtax à partir du composant `dynamicForm` qui permet de créer dynamiquement un formulaire en déclarant ses champs et leur type (#318)
- Amélioration du composant de recherche d'un taxon en ne recherchant que sur les débuts de mot et en affichant en premier les noms de référence (`order_by cd_nom=cd_ref DESC`) #334
- Multilingue fourni avec français et anglais (extensible à d'autres langues)

- Mise en place d'un export CSV, SHP, GeoJSON paramétrable dans Occtax. #363 et #366
- Ajout d'un message d'erreur si l'utilisateur n'a pas de jeu de données ou si il y a eu un problème lors de la récupération des JDD depuis MTD
- Prise en compte du CRUVED au niveau des routes et du front pour adapter les contenus et fonctionnalités aux droits de l'utilisateur
- Mise en place des triggers alimentant la synthèse à partir des données saisies et modifiées dans Occtax

### Synthèse

Module permettant de rechercher parmi les données des différentes sources présentes ou intégrées dans la base de données de GeoNature

- Mise en place du backend, de l'API et du frontend #345
- Interface de consultation, de recherche et d'export dans la Synthèse
- Synthèse : Calcul automatique (trigger) des zonages de chaque observation (communes, zonages réglementaires et naturels)
- Recherche sur les zonages générique et paramétrable
- Recherche par taxon, liste de taxons, par rang, groupe, liste rouge, milieu, attribut taxonomique, nomenclature, date, période, commune, zonage, cadre d'acquisition, jeu de données, observateur, polygone, rectangle ou cercle dessiné
- Retour à la fiche source possible si l'observation a été saisie dans un module de GeoNature
- Affichage de la fiche détail de chaque observation
- Attributs TaxHub dynamiques et paramétrables
- Configuration possible du module (colonnes, limites de recherche et d'export, zoom, export. . . )
- Export basé sur une vue (observations et statuts)
- Prise en compte du CRUVED pour définir les données à afficher et à exporter #412
- Recherche de taxons : Liste basée sur une table alimentée automatiquement par les taxons présents au moins une fois dans la Synthèse

### Export

Module permettant de proposer des exports basés sur des vues

- Mise en place temporaire d'un export unique, basé sur une vue s'appuyant sur les données de Occtax, par jeu de données
- A remplacer par le module générique [https://github.com/PnX-SI/gn\\_module\\_export](https://github.com/PnX-SI/gn_module_export) (en cours de développement) permettant de générer des exports à volonté en créant des vues et en les affectant à des utilisateurs ou des groupes. Chaque export sera accompagné de son API standardisée et documentée

### Admin

Module d'administration des tables centrales de GeoNature

- Mise en place d'un module (incomplet) permettant de gérer les métadonnées et les nomenclatures

### Gestion des droits

- Mise en place d'un système baptisé CRUVED permettant de définir globalement ou par module 6 actions sont possibles (Create / Read / Update / Validate / Export / Delete) sur 3 portées possibles (Mes données / Les données de mon organisme / Toutes les données)

- Ces évolutions ont été intégrées au niveau du schéma `utilisateurs` de la base de données de UsersHub, de son module (<https://github.com/PnX-SI/UsersHub-authentication-module>), des routes de l'API GeoNature et des interfaces

### Bases de données

- Développement d'un module et d'une API générique et autonome pour la gestion des nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module>). Il permet d'avoir un mécanisme générique de centralisation des listes de valeurs (nomenclatures) pour ne pas créer des tables pour chaque liste : <https://github.com/PnX-SI/Nomenclature-api-module>. Les valeurs de chaque nomenclature s'adaptent en fonction des regnes et groupe 2 INPN des taxons.
- Mise en place de tables de stockage verticales (historique, médias et validation) #339
- Mise en place d'un référentiel géographique avec un schéma dédié (`ref_geo`), partageable avec d'autres applications comprenant une table des communes, une table générique des zonages, une table pour le MNT et des fonctions pour intersecter point/ligne/polygones avec les zonages et le MNT (#228)
- Evolution du schéma `utilisateurs` de UsersHub pour passer d'une gestion des droits avec 6 niveaux à un mécanisme plus générique, souple et complet. Il permet d'attribuer des actions possibles à un rôle (utilisateur ou groupe), sur une portée; dans une application ou un module. 6 actions sont possibles dans GeoNature : Create / Read / Update / Validate / Export / Delete (aka CRUVED). 3 portées de ces actions sont possibles : Mes données / Les données de mon organisme / Toutes les données.
- Droits CRUVED : La définition du CRUVED d'un rôle (utilisateur ou groupe) sur un module de GeoNature surcouche ses droits GeoNature même si ils sont inférieurs. Si une action du CRUVED n'est pas définie au niveau du module, on prend celle de l'application parente. #292
- Si un rôle a un R du CRUVED à 0 pour un module, alors celui-ci ne lui est pas listé dans le Menu et il ne lui est pas accessible si il en connaît l'URL. #360
- Développement des métadonnées dans la BDD (schéma `gn_meta`) sur la base du standard Métadonnées du SINP (<http://standards-sinp.mnhn.fr/category/standards/metadonnees/>). Elles permettent de gérer des jeux de données, des cadres d'acquisition, des acteurs (propriétaire, financeur, producteur...) et des protocoles. Chaque relevé est associé à un jeu de données.
- Développement d'un mécanisme de calcul automatique de la sensibilité d'une espèce directement dans la BDD (sur la base des règles nationales et régionales du SINP + locales éventuellement)
- Intégration du calcul automatique de l'identifiant permanent SINP (#209)
- Création du schéma `gn_monitoring` pour gérer la partie générique des modules de suivi (sites et visites centralisés) et les routes associées
- Mise en place d'un schéma `gn_commons` dans la BDD qui permet de stocker de manière générique des informations qui peuvent être communes aux autres modules : l'historique des actions sur chaque objet de la BDD, la validation d'une donnée et les médias associés à une donnée. Accompagné de fonctions génériques d'historisation et de validation des données mises en place sur le module Occtax. #339
- Ajout d'une vue matérialisée (`gn_synthese.vm_min_max_for_taxons`) et d'une fonction (`gn_synthese.fct_calculate_min_max_for_taxon`) permettant de renvoyer des informations sur les observations existantes d'un taxon (étendue des observations, date min et max, altitude min et max, nombre d'observations) pour orienter la validation et la saisie ([https://github.com/PnX-SI/gn\\_module\\_validation/issues/5](https://github.com/PnX-SI/gn_module_validation/issues/5)). Désactivée pour le moment.
- Ajout d'un trigger générique pour calculer la géométrie dans la projection locale à partir de la géométrie 4326 (#370)
- Ajout d'un trigger pour calculer automatiquement les zonages des sites de suivi (`gn_monitoring.fct_trg_cor_site_area()`)

- Gestion des conflits de nomenclatures en n'utilisant plus leur `id_type` ni leur `id_nomenclature` lors de la création de leur contenu (`code_nomenclature`) (#384)
- Mise en place d'un schéma `gn_imports` intégrant des fonctions SQL permettant d'importer un CSV dans la BDD et de mapper des champs de tables importées avec ceux d'une table de GeoNature pour générer le script `INSERT INTO`
- Début de script de migration GeoNature V1 vers GeoNature V2
- Nombreuses fonctions intégrées dans les schémas de la BDD

### Installation

- Scripts d'installation autonome ou globale de GeoNature sur Debian (8 et 9) et Ubuntu (16 et 18)
- Scripts de déploiement spécifiques de DEPOBIO (MTES-MNHN)

### Documentation

- Rédaction d'une documentation concernant l'installation (autonome ou globale), l'utilisation, l'administration et le développement : <http://docs.geonature.fr>

### Développement

- Découpage de l'application en backend / API / Frontend
- Multilingue au niveau de l'interface et des listes de valeurs avec français et anglais intégrés mais extensible à d'autres langues (#173)
- Développement de composants Angular génériques pour pouvoir les utiliser dans plusieurs modules sans avoir à les redévelopper ni les dupliquer (composant CARTE, composant RECHERCHE TAXON, composant OBSERVATEURS, composant NOMENCLATURES, SelectSearch, Municipalities, Observers, DynamicForm, MapList...)
- Implémentation de la gestion des droits au niveau de l'API (pour limiter les données affichées à un utilisateur en fonction de ses droits) et au niveau du Frontend (pour afficher ou non certains boutons aux utilisateurs en fonction de leurs droits).
- Par défaut, l'authentification et les utilisateurs sont gérés localement dans UsersHub, mais il est aussi possible de connecter GeoNature directement au CAS de l'INPN, sans UsersHub (cas de l'instance nationale INPN de GeoNature).
- Connexion possible au webservice METADONNEES de l'INPN pour y récupérer les jeux de données en fonction de l'utilisateur connecté, avec mise à jour des JDD à chaque appel de la route
- Mise en place d'un mécanisme standardisé de développement de modules dans GeoNature (#306)
- Ajout de tests unitaires au niveau du backend et du frontend
- Ajout d'un mécanisme de log par email (paramètres MAILERROR)
- Début de création du module de gestion des médias (backend uniquement)
- Mise en place d'une configuration globale et d'une configuration par module
- Fonction d'installation d'un module et de génération des fichiers de configuration
- Gestion de l'installation d'un module qui n'a pas de Frontend dans GeoNature
- Mise en place d'une route générique permettant de requêter dans une vue non mappée
- Mise en place d'un script pour la customisation de la plateforme nationale ([https://github.com/PnX-SI/GeoNature/blob/develop/install\\_all/configuration\\_mtes.sh](https://github.com/PnX-SI/GeoNature/blob/develop/install_all/configuration_mtes.sh))

### Autres modules

- Module Export en cours de développement ([https://github.com/PnX-SI/gn\\_module\\_export](https://github.com/PnX-SI/gn_module_export)). Chaque export s'appuie sur une vue. Il sera possible aux administrateurs d'une GeoNature d'ajouter autant de vues que nécessaires dans son GeoNature.
- Module de validation des données en cours de développement ([https://github.com/PnX-SI/gn\\_module\\_validation/issues/4](https://github.com/PnX-SI/gn_module_validation/issues/4))
- Module Suivi Flore territoire en cours de développement ([https://github.com/PnX-SI/gn\\_module\\_suivi\\_flore\\_territoire](https://github.com/PnX-SI/gn_module_suivi_flore_territoire))
- Module Suivi Habitat en cours de développement ([https://github.com/PnX-SI/gn\\_module\\_suivi\\_habitat\\_territoire/issues/1](https://github.com/PnX-SI/gn_module_suivi_habitat_territoire/issues/1))
- gn\_module\_suivi\_chiro refondu pour devenir un module de GeoNature V2 ([https://github.com/PnCevennes/gn\\_module\\_suivi\\_chiro](https://github.com/PnCevennes/gn_module_suivi_chiro))
- Projet suivi utilisé comme Frontend générique et autonome pour le Suivi chiro ([https://github.com/PnCevennes/projet\\_suivis\\_frontend](https://github.com/PnCevennes/projet_suivis_frontend))
- GeoNature-citizen en cours de développement (<https://github.com/PnX-SI/GeoNature-citizen/issues/2>)
- GeoNature-mobile en cours de refonte pour compatibilité avec GeoNature V2 (<https://github.com/PnEcrins/GeoNature-mobile/issues/19>)
- GeoNature-atlas en cours d'ajustements pour compatibilité avec GeoNature V2 (<https://github.com/PnX-SI/GeoNature-atlas/issues/162>)

### Notes de version

#### 1. Pour les utilisateurs utilisant la version 1 de GeoNature :

Il ne s'agit pas de mettre à jour GeoNature mais d'en installer une nouvelle version. En effet, il s'agit d'une refonte complète.

- Passer à la dernière version 1 de GeoNature (1.9.1)
- Idem pour UsersHub et TaxHub
- Installer GeoNature standalone ou refaire une installation complète
- Adaptez les scripts présents dans `/data/migrations/v1tov2` et exécutez-les

*TODO : MAJ depuis V1 à tester et compléter*

#### 2. Pour les utilisateurs utilisant la version 2.0.0.beta5 :

- Supprimer le schéma `gn_synthese` puis le recréer dans sa version RC1 (#430)
- Exécuter l'update de la BDD GeoNature (`data/migrations/2.0.0beta5-to-2.0.0rc1.sql`) ainsi que celui du sous-module Nomenclature (<https://github.com/PnX-SI/Nomenclature-api-module/blob/1.2.1/data/update1.1.0to1.2.1.sql>)
- Suivre la procédure habituelle de mise à jour
- Exécuter les commandes suivantes :

```
cd geonature/backend
source venv/bin/activate
geonature generate_frontend_modules_route
geonature frontend_build
```

## 10.41 2.0.0.beta5 (2018-07-16)

### Nouveautés

- Ajout d'un message d'erreur si l'utilisateur n'a pas de JDD ou si il y a eu un problème lors de la récupération des JDD de MTD
- Ajout d'une vue matérialisée (`gn_synthese.vm_min_max_for_taxons`) et d'une fonction (`gn_synthese.fct_calculate_min_max_for_taxon`) permettant de renvoyer des informations sur les observations existantes d'un taxon (étendue des observations, date min et max, altitude min et max, nombre d'observations) pour orienter la validation et la saisie ([https://github.com/PnX-SI/gn\\_module\\_validation/issues/5](https://github.com/PnX-SI/gn_module_validation/issues/5))
- L'export OccTax est désormais basé sur une vue qu'il est possible d'adapter
- Ajouts de nouveaux tests automatisés du code et mise en place de Travis pour les lancer automatiquement à chaque commit (<https://travis-ci.org/PnX-SI/GeoNature>)
- Ajout de données test
- Mise à jour des scripts de déploiement spécifiques de DEPOBIO (MTES)
- Déplacement de la table centrale de gestion des paramètres `t_parameters` dans le schéma `gn_commons` (#376)
- Ajout d'un trigger générique pour calculer la géométrie dans la projection locale à partir de la géométrie 4326 (#370)
- Regroupement des fichiers liés à l'installation et la mise à jour dans un répertoire dédié (`install`) (#383)
- Mise en place de scripts de migration global de la BDD (`data/migrations/2.0.0beta4to2.0.0beta5.sql`) et du schéma `pr_occtax` (`contrib/occtax/data/migration_2.0.0beta4to2.0.0beta5.sql`), d'un script générique de migration de l'application (`install/migration/migration.sh`) et d'une doc de mise à jour (<https://github.com/PnX-SI/GeoNature/blob/develop/docs/installation-standalone.rst#mise-%C3%A0-jour-de-lapplication>)
- Réintégration des fichiers de configuration, de logs et des modules externes dans les répertoires de l'application (#375)
- Ajout de routes à `gn_monitoring`
- Ajout d'un trigger pour calculer automatiquement les zonages des sites de suivi (`gn_monitoring.fct_trg_cor_site_area()`)
- Améliorations et documentation des commandes d'installation d'un module
- Ajout des unités géographiques dans le schéma `ref_geo`
- Ajout d'un bouton Annuler dans le formulaire Occtax
- Gestion des conflits de nomenclatures en n'utilisant plus leur `id_type` ni leur `id_nomenclature` (#384)
- Migration du SQL de `ref_nomenclautres` dans le dépôt du sous-module (<https://github.com/PnX-SI/Nomenclature-api-module>)
- Début de mise en place d'un backoffice (métadonnées et nomenclatures)

### Corrections

- OccTax : Correction du double post
- OccTax : Correction des droits dans les JDD
- OccTax : Correction de l'affichage des `observers_txt` dans la fiche d'un relevé



- Correction de la gestion générique des médias
- Suppression du lien entre `ref_geo` et `ref_nomenclatures` (#374)
- Compléments et relecture de la documentation
- Correction

### Notes de version

Si vous mettez à jour votre GeoNature depuis une Beta4 :

- Téléchargez la beta5 et renommez les répertoires :

```
cd /home/myuser
wget https://github.com/PnX-SI/GeoNature/archive/geonature2beta.zip
unzip geonature2beta.zip
mv /home/<mon_user>/geonature/ /home/<mon_user>/geonature_old/
mv GeoNature-geonature2beta /home/<mon_user>/geonature/
```

- Exécutez le script de migration `install/migration/beta4tobeta5.sh` depuis la racine de votre GeoNature :

```
cd geonature
./install/migration/beta4tobeta5.sh
```

Celui-ci va récupérer vos fichiers de configuration, déplacer les modules et appliquer les changements de la BDD.

- Si vous avez développé des modules externes, voir <https://github.com/PnX-SI/GeoNature/issues/375>, en ajoutant un lien symbolique depuis le répertoire `external_modules` et en réintégrant la configuration du module dans son répertoire `config`

## 10.42 2.0.0.beta4 (2018-05-25)

### Nouveautés

- Synthèse : début de mise en place du backend, de l'API et du frontend #345
- Complément de la nomenclature des Méthodes de détermination et suppression du champs `Complement_Determination`. Merci @DonovanMaillard. #341
- Nouveaux composants Angular (SelectSearch, Municipalities, Observers)
- Amélioration de composants Angular (Date du jour par défaut, Option de tri des nomenclatures, DynamicForm)
- Connexion à MTD INPN : Mise à jour des JDD à chaque appel de la route
- Finalisation du renommage de Contact en OccTax (BDD, API, backend)
- Droits CRUVED : La définition du CRUVED d'un rôle (utilisateur ou groupe) sur un module de GeoNature surcoud ses droits GeoNature même si ils sont inférieurs. Si une action du CRUVED n'est pas définie au niveau du module, on prend celle de l'application parente. #292
- Si un rôle a un R du CRUVED à 0 pour un module, alors celui-ci ne lui est pas listé dans le Menu et il ne lui est pas accessible si il en connaît l'URL. #360
- Mise en place d'un schéma `gn_commons` dans la BDD qui permet de stocker de manière générique des informations qui peuvent être communes aux autres modules : l'historique des actions sur chaque objet de la BDD, la validation d'une donnée et les médias associés à une donnée. Accompagné de fonctions génériques d'historisation et de validation des données mises en place sur le module OccTax. #339
- Amélioration de l'ergonomie du MapList de OccTax. #361



- Mise en place d'un export CSV, SHP, GeoJSON paramétrable dans OccTax. #363 et #366
- Amélioration du module générique `gn_monitoring` et de ses sous-modules [https://github.com/PnCevennes/gn\\_module\\_suivi\\_chiro](https://github.com/PnCevennes/gn_module_suivi_chiro) et [https://github.com/PnCevennes/projet\\_suivis\\_frontend](https://github.com/PnCevennes/projet_suivis_frontend)
- Amélioration et compléments des scripts d'installation
- Mise en place d'un script pour la customisation de la plateforme nationale ([https://github.com/PnX-SI/GeoNature/blob/develop/install\\_all/configuration\\_mtes.sh](https://github.com/PnX-SI/GeoNature/blob/develop/install_all/configuration_mtes.sh))

### Documentation

- Complément des différentes documentations
- Ajout d'une documentation d'administration d'OccTax (<https://github.com/PnX-SI/GeoNature/blob/develop/docs/admin-manual.rst#module-occtax>)

## 10.43 2.0.0.beta3 (2018-03-28)

### Nouveautés

- Travail sur le module générique de Suivi intégré à GeoNature (`gn_monitoring`). Gestion des fichiers de configuration
- Gestion de l'installation d'un module qui n'a pas de Frontend dans GeoNature
- Mise en place de tests automatiques au niveau du Frontend
- Ménage et réorganisation du code du Frontend
- Factorisation et harmonisation des composants génériques Angular
- Suppression des blocs non fonctionnels sur la Home
- Mise à jour de la doc et du MCD
- Possibilité de masquer ou afficher les différents champs dans le formulaire Occtax (#344)
- Ajout des nomenclatures dans les filtres d'OccTax à partir du nouveau composant `dynamicForm` qui permet de créer dynamiquement un formulaire en déclarant les champs (#318)
- Amélioration du composant de recherche d'un taxon en ne recherchant que sur les débuts de mot et en affichant en premier les noms de référence (`order_by cd_nom=cd_ref DESC`) - #334
- Mise en place d'une route générique permettant de requêter dans une vue non mappée
- Suppression des options vides dans les listes déroulantes des nomenclatures
- Ajout de quelques paramètres (niveau de zoom mini dans chaque module, ID de la liste des taxons saisissables dans Occtax...)

### Corrections

- Correction de la pagination du composant `MapList`
- Correction des droits attribués automatiquement quand on se connecte avec le CAS
- Correction de l'installation optionnelle de UsersHub dans le script `install_all.sh`

### Modules annexes

- Début de refonte du module Suivi chiro ([https://github.com/PnCevennes/gn\\_module\\_suivi\\_chiro](https://github.com/PnCevennes/gn_module_suivi_chiro)) connecté au module générique de suivi de GeoNature, dont le front sera externe à GeoNature ([https://github.com/PnCevennes/projet\\_suivi](https://github.com/PnCevennes/projet_suivi))

- Maquettage et avancée sur le module Validation ([https://github.com/PnX-SI/gn\\_module\\_validation](https://github.com/PnX-SI/gn_module_validation))
- Définition du module Suivi Habitat Territoire ([https://github.com/PnX-SI/gn\\_module\\_suivi\\_habitat\\_territoire](https://github.com/PnX-SI/gn_module_suivi_habitat_territoire))
- Piste de définition du module Interopérabilité ([https://github.com/PnX-SI/gn\\_module\\_interoperabilite](https://github.com/PnX-SI/gn_module_interoperabilite))

## 10.44 2.0.0.beta2 (2018-03-16)

### Nouveautés

- Compléments de la documentation (schéma architecture, administration, installation, développement, FAQ...)
- Amélioration de l'ergonomie du module OccTax (composant MapList, filtres, colonnes et formulaires) et du module Exports
- Amélioration du composant de recherche d'un taxon (#324)
- Amélioration et optimisation de la sérialisation des données
- Ajout de tests unitaires au niveau du backend
- Ajout d'un mécanisme de log par email (paramètres MAILERROR)
- Migration du module occtax dans le répertoire `/contrib` pour homogénéiser les modules
- Création du schéma `gn_monitoring` pour gérer la partie générique des modules de suivi (sites et visites centralisés)
- Début de création du module générique des protocoles de suivi
- Début de création du module de gestion des médias

### Corrections

- Corrections de l'installation globale et autonome
- Renommage Contact en OccTax (en cours)
- Nettoyage du schéma des métadonnées (`gn_meta`)

## 10.45 2.0.0.beta1 (2018-02-16)

La version 2 de GeoNature est une refonte complète de l'application.

- Refonte technologique en migrant de PHP/Symfony/ExtJS/Openlayers à Python3/Flask/Angular4/Leaflet
- Refonte de l'architecture du code pour rendre GeoNature plus générique et modulaire
- Refonte de la base de données pour la rendre plus standard, plus générique et modulaire
- Refonte ergonomique pour moderniser l'application

Présentation et suivi du projet : <https://github.com/PnX-SI/GeoNature/issues/168>

### Nouveautés

- Refonte de la base de données du module Contact, renommé en OccTax, s'appuyant sur le standard Occurrence de taxons du SINP (#183)
- Développement du module OccTax regroupant les contacts Faune, Flore, Fonge et Mortalité (avec formulaire de consultation et de saisie des données)

- Développement d'un module et d'une API générique et autonome pour la gestion des nomenclatures (<https://github.com/PnX-SI/Nomenclature-api-module>). Il permet d'avoir un mécanisme générique de centralisation des listes de valeurs (nomenclatures) pour ne pas créer des tables pour chaque liste : <https://github.com/PnX-SI/Nomenclature-api-module>. Les valeurs de chaque nomenclature s'adaptent en fonction des regnes et groupe 2 INPN des taxons.
- Découpage de l'application en backend / API / Frontend
- Multilingue au niveau de l'interface et des listes de valeurs avec français et anglais intégrés mais extensible à d'autres langues (#173)
- Développement de composants génériques pour pouvoir les utiliser dans plusieurs modules sans avoir à les redévelopper ni les dupliquer (composant CARTE, composant RECHERCHE TAXON, composant OBSERVATEURS, composant NOMENCLATURES...)
- Mise en place d'un référentiel géographique avec un schéma dédié (`ref_geo`), partageable avec d'autres applications comprenant une table des communes, une table générique des zonages, une table pour le MNT et des fonctions pour intersecter point/ligne/polygones avec les zonages et le MNT (#228)
- Evolution du schéma `utilisateurs` de UsersHub pour passer d'une gestion des droits avec 6 niveaux à un mécanisme plus générique, souple et complet. Il permet d'attribuer des actions possibles à un rôle (utilisateur ou groupe), sur une portée; dans une application ou un module. 6 actions sont possibles dans GeoNature : Create / Read / Update / Validate / Export / Delete (aka CRUVED). 3 portées de ces actions sont possibles : Mes données / Les données de mon organisme / Toutes les données.
- Implémentation de la gestion des droits au niveau de l'API (pour limiter les données affichées à un utilisateur en fonction de ses droits) et au niveau du Frontend (pour afficher ou non certains boutons aux utilisateurs en fonction de leurs droits).
- Par défaut, l'authentification et les utilisateurs sont gérés localement dans UsersHub, mais il est aussi possible de connecter GeoNature au CAS de l'INPN, sans utiliser GeoNature (utilisé pour l'instance nationale INPN de GeoNature). GeoNature peut aussi se connecter au webservice METADONNEES de l'INPN pour y récupérer les jeux de données en fonction de l'utilisateur connecté.
- Mise en place d'un module d'export. Chaque export s'appuie sur une vue. Il sera possible à chaque administrateur d'ajouter autant de vues que nécessaires dans son GeoNature. Pour le moment, un export au format SINP Occurrence de taxons a été intégré par défaut.
- Développement des métadonnées dans la BDD (schema `gn_meta`) sur la base du standard Métadonnées du SINP (<http://standards-sinp.mnhn.fr/category/standards/metadonnees/>). Elles permettent de gérer des jeux de données, des cadres d'acquisition, des acteurs (propriétaire, financeur, producteur...) et des protocoles. Chaque relevé est associé à un jeu de données.
- Développement d'un mécanisme de calcul automatique de la sensibilité d'une espèce directement dans la BDD (sur la base des règles nationales et régionales du SINP + locales éventuellement)
- Intégration du calcul automatique de l'identifiant permanent SINP (#209)
- Mise en place d'un mécanisme standardisé de développement de modules dans GeoNature (#306)
- Scripts d'installation autonome ou globale de GeoNature sur Debian 8 et 9

## Documentation

- Installation globale de GeoNature (avec TaxHub et UsersHub) / <https://github.com/PnX-SI/GeoNature/blob/develop/docs/installation-all.rst>
- Installation autonome de GeoNature / <https://github.com/PnX-SI/GeoNature/blob/develop/docs/installation-standalone.rst>
- Manuel utilisateur / <https://github.com/PnX-SI/GeoNature/blob/develop/docs/user-manual.rst>
- Manuel administrateur / <https://github.com/PnX-SI/GeoNature/blob/develop/docs/admin-manual.rst>

- Développement (API, modules et composants) / <https://github.com/PnX-SI/GeoNature/blob/develop/docs/development.rst>

Documentation complète disponible sur <http://geonature.fr/docs/2-0-0-beta1>

#### A venir

- Finalisation MCD du module Synthèse
- Triggers d'alimentation automatique de la Synthèse depuis le module OccTax
- Développement de l'interface du module Synthèse
- Amélioration et généricité du module OccTax (médias, import GPX, champs masquables et pseudo-champs)
- Généricité du module d'export
- Développement du module de validation (#181)
- Développement d'un module de suivi des habitats avec une gestion générique des sites et visites de suivi
- Développement d'un module de collecte citoyenne (#242)

## 10.46 1.9.1 (2018-05-17)

#### Corrections

- Installation - Suppression des couches SIG (communes, znieff...) pour les télécharger sur <http://geonature.fr/data/inpn/layers/> et ainsi alléger le dépôt de 158 Mo.
- Compléments mineurs de la documentation
- Migration du script `install_all` en Debian 9. La doc et le script Debian 8 restent disponibles dans le répertoire `docs/install_all`
- Corrections mineures de triggers
- Compatibilité avec TaxHub 1.3.2, UsersHub 1.3.1, GeoNature-atlas 1.3.2

#### Notes de version

- Vous pouvez passer directement d'une 1.7.X à la 1.9.1, en prenant en compte les notes des différentes versions intermédiaires, notamment les scripts de mise à jour de la BDD ainsi que les éventuels nouveaux paramètres à ajouter.
- Exécuter le script de mise à jour de la BDD `data/update_1.9.0to1.9.1.sql`

## 10.47 1.9.0 (2017-07-06)

**ATTENTION : Les évolutions de cette version concernent aussi la webapi. Si vous utilisez les applications GeoNature-mobile, vous devez attendre la sortie d'une version de GeoNature-mobile-webapi (<https://github.com/PnEcrins/GeoNature-mobile-webapi>) compatible avec cette version 1.9.0 de GeoNature. Coming soon !**

A noter aussi que cette version de GeoNature est compatible avec GeoNature-atlas 1.2.4 et +.

#### Nouveautés

- Ajout de la création des index spatiaux à la création initiale de la base.
- Création ou mise à jour des géométries compatible PostGIS 2.

- Ajout du champ diffusable (oui/non) dans le formulaire web de saisie, uniquement pour ContactFaune et Mortalité (TODO : faire la même chose pour les autres protocoles).
- Multi-projection : Les versions antérieures de GeoNature n'étaient compatibles qu'avec la projection Lambert 93 (srid: 2154). Cette version permet de choisir sa projection locale. Elle ajoute un paramètre `srid_local` dans le `config/settings.ini` et renomme tous les champs `the_geom_2154` en `the_geom_local` des tables "métier". Ce paramètre est notamment utilisé lors de la création de la base pour affecter le srid de la projection locale à tous les champs `the_geom_local` présents dans les tables de la base. Ce paramètre est également utilisé pour mettre en cohérence le système de projection local utilisé dans toutes les couches SIG présentes dans la base et les géométries stockées dans les champs `the_geom_local` des tables "métier". Le paramétrage du service WMS dans `wms/wms.map` est également pris en charge par le script d'installation de l'application.
- Correction de l'installation de npm
- Script `install_all.sh` mis à jour avec les nouvelles versions de GeoNature-atlas, de TaxHub et de UserHub.

**IMPORTANT :** toutes les couches SIG insérées dans le schéma `layers` doivent être dans la projection fournie pour le paramètre `srid_local`. L'application est livrée avec un ensemble de couches en Lambert 93 concernant la métropole. Une installation avec une autre projection, hors métropole, doit donc se faire sans l'insertion des couches SIG. Vous devrez manuellement fournir le contenu des tables du schéma `layers` dans la projection choisie.

#### Notes de versions

- Vous pouvez ajouter les paramètres `srid_local`, `install_sig_layers` et `add_sample_data` au fichier `config/settings.ini` en vous inspirant du fichier `config/settings.ini.sample`. Toutefois ces paramètres ne sont utilisés que pour une nouvelle installation et notamment pour l'installation de la base.
- Vous pouvez passer directement d'une 1.7.X à la 1.9.0, en prenant en compte les notes des différentes versions intermédiaires, notamment les scripts de mise à jour de la BDD ainsi que les éventuels nouveaux paramètres à ajouter.
- Si vous migrez depuis la version 1.8.3, exécutez le fichier SQL `data/update_1.8.3to1.9.0.sql`. Comme GeoNature ne fonctionne jusque là que pour des structures de métropole, il est basé sur le fait que le champ `the_geom_local` reste en Lambert 93 (2154). Assurez-vous que le paramètre `$srid_local` dans `lib/sfGeonatureConfig.php` est égal à 2154. **ATTENTION :** ce script SQL renomme tous les champs `the_geom_2154` en `the_geom_local` de la BDD de GeoNature. Ceci affecte de nombreuses tables, de nombreux triggers et de nombreuses vues de la base. Le script n'intègre que les vues fournies par défaut. Si vous avez créé des vues spécifiques, notamment pour le module d'export, ou si vous avez modifié des vues fournies, vous devez adapter/compléter le script. Vous pouvez vous inspirer de son contenu.
- **RAPPEL :** Ceci affecte également la webapi des applications mobiles. Vous devez donc mettre à jour votre webapi si vous utilisez la saisie sur les applications mobiles. Une release de la webapi devrait sortir bientôt.

## 10.48 1.8.4 (2017-04-10)

### Corrections

- Correction du script d'installation globale (`install_all`) si l'utilisateur de BDD par défaut a été renommé (`data/grant.sql`)
- Correction de la création des vues qui remontent la liste des taxons dans les 3 contacts

## 10.49 1.8.3 (2017-02-23)

### Nouveautés

- Multi-organisme : l'organisme associé à la donnée est désormais celui de l'utilisateur connecté dans l'application (lors de la création d'une observation uniquement).
- Taxonomie : création d'une liste Saisie possible, remplaçant l'attribut Saisie. Cela permet de choisir les synonymes que l'on peut saisir ou non dans GeoNature en se basant sur les `cd_nom` (`bib_listes` et `cor_nom_liste`) et non plus sur les `cd_ref` (`bib_attributs` et `cor_taxon_attribut`). Voir le script de migration SQL `data/update_1.8.2to1.8.3.sql` pour bien basculer les informations de l'attribut dans la nouvelle liste.
- Correction de la vue `synthese.v_tree_taxons_synthese` potentiellement bloquante à l'ouverture de la synthèse.
- Suppression de la table `utilisateurs.bib_observateurs` inutile.
- Création des index spatiaux manquants (performances)
- Clarification et corrections mineures du script `install_all`
- Ajout du MCD de la 1.8 (par @xavier-pnm)
- Améliorations du nom des fichiers exportés depuis la Synthèse (par @sylvain-m)

### Notes de versions

Vous pouvez supprimer les lignes concernant le paramètre `public static $id_organisme = ...` dans `lib/sfGeonatureConfig.php`, l'organisme n'étant plus un paramètre fixe mais désormais celui de l'utilisateur connecté.

Vous pouvez passer directement d'une 1.7.X à la 1.8.3, en prenant en compte les notes des différentes versions intermédiaires.

Si vous migrez depuis la version 1.8.2, exécutez le fichier SQL `data/update_1.8.2to1.8.3.sql`.

## 10.50 1.8.2 (2017-01-11)

### Nouveautés

- Modularité des scripts SQL de création de la base en les dissociant par protocole et en regroupant les triggers dans les schémas de chaque protocole (préparation GeoNature V2)
- Correction d'une requête dans flore station (indépendance vis à vis de flore patrimoniale)
- Correction du trigger `synthese_update_fiche_cflore` (@ClaireLagaye)

### Notes de versions

Vous pouvez passer directement d'une 1.7.X à la 1.8.2, en prenant en compte les notes des différentes versions intermédiaires.

Si vous migrez depuis la version 1.8.1, exécutez le fichier `data/update_1.8.1to1.8.2.sql`. Consultez les dernières lignes de ce fichier : vous devez évaluer si la requête d'insertion dans la table `taxonomie.cor_taxon_attribut` doit être faite ou non (vous pourriez avoir déjà constaté et corrigé cette erreur lors d'une précédente migration). Cela corrige l'absence de taxons protégés dans votre synthèse en récupérant les informations de protection présentes dans le champ `filtre3` de la table `save.bib_taxons`

## 10.51 1.8.1 (2017-01-05)

### Nouveautés

- Ajout des sauvegardes et de l'installation globale avec un exemple détaillé dans la documentation : <http://docs.geonature.fr>
- Optimisation et correction de la vue qui retourne l'arbre des rangs taxonomiques (`synthese.v_tree_taxons_synthese`)
- Mise en cohérence des données exemple de GeoNature-atlas avec les critères des vues matérialisées de GeoNature-atlas
- Mise à jour de 2 triggers du Contact Flore (@ClaireLagaye)

### Notes de versions

Vous pouvez passer directement d'une 1.7.X à la 1.8.1, en prenant en compte les notes des différentes versions intermédiaires.

Si vous migrez depuis la version 1.8.0, exécutez le fichier `data/update_1.8to1.8.1.sql`

## 10.52 1.8.0 (2016-12-14)

### Nouveautés

- Passage à TAXREF version 9
- Accès à la synthèse en consultation uniquement pour des utilisateurs enregistrés avec des droits 1
- Ajout d'un champ `diffusion` (oui/non) dans la table `synthese.syntheseeff`, utilisable dans GeoNature-atlas. Pas d'interface de gestion de ce champ pour le moment. CF #132
- Création d'un script d'installation simplifié pour un pack UsersHub, TaxHub, GeoNature et GeoNature-atlas : [https://github.com/PnX-SI/GeoNature/tree/master/docs/install\\_all](https://github.com/PnX-SI/GeoNature/tree/master/docs/install_all)
- Factorisation des SQL de création des schémas `taxonomie` et `utilisateurs` en les récupérant dans les dépôts TaxHub et UsersHub
- Compatibilité avec l'application [TaxHub](#) qui permet de gérer la taxonomie à partir de TAXREF. Cela induit d'importants changements dans le schéma `taxonomie`, notamment le renommage de `taxonomie.bib_taxons` en `taxonomie.bib_noms`, la suppression de `taxonomie.bib_filtres` et l'utilisation de `taxonomie.bib_attributs` (voir <https://github.com/PnX-SI/TaxHub/issues/71> pour plus d'informations). Voir aussi le fichier de migration `data/update_1.7to1.8.sql` qui permet d'automatiser ces évolutions de la BDD
- Compatibilité avec l'application [GeoNature-atlas](#) qui permet de diffuser les données de la synthèse faune et flore dans un atlas en ligne (exemple : <http://biodiversite.ecrins-parcnational.fr>)
- Création d'un site internet de présentation de GeoNature : <http://geonature.fr>

### Corrections

- Amélioration des triggers concernant la suppression de fiches orphelines
- Affichage par défaut du nom latin dans Contact flore et Contact invertébrés
- Correction des exports lors de la présence de points-virgules dans les commentaires. Fix #143
- Suppression du besoin d'un super utilisateur lors de l'installation de la BDD. Fix #141
- Correction de l'ID des protocoles mortalité et invertébrés dans la configuration par défaut

- Suppression d'un doublon dans le fichier de configuration symfony de l'application
- Correction des coordonnées lors de l'export de données Flore Station
- Autres corrections mineures

#### Note de version

- Exécuter le script SQL de migration réalisant les modifications de la BDD de la version 1.7.X à 1.8.0 `data/update_1.7to1.8.sql`
- Mettre à jour taxref en V9 en vous inspirant du script `data/taxonomie/inpn/update_taxref_v8tov9`

#### TaxHub

L'application TaxHub (<https://github.com/PnX-SI/TaxHub>) est désormais fonctionnelle, documenté et installable.

Elle vous aidera à gérer vos taxons et l'ensemble du schéma `taxonomie`, présent dans la BDD de GeoNature.

TaxHub évoluera pour intégrer progressivement de nouvelles fonctionnalités.

Il est conseillé de ne pas installer la base de données de TaxHub indépendamment et de connecter l'application directement sur la base de données de GeoNature.

#### GeoNature-atlas

GeoNature-atlas est également basé sur le schéma `taxonomie` de TaxHub. Ainsi TaxHub permet la saisie des informations relatives aux taxons (descriptions, milieux, photos, liens, PDF...). GeoNature-atlas dispose de sa propre base de données mais pour fonctionner en connexion avec le contenu de la base GeoNature il faut à minima disposer d'une version 1.8 de GeoNature.

**notes** Une régression dans le contenu de Taxref V9 conduit à la suppression de l'information concernant le niveau de protection des espèces (régional, national, international...). Cette information était utilisée par GeoNature, notamment pour définir les textes à retenir pour la colonne `concerne_mon_territoire` de la table `taxonomie`. `taxref_protection_articles`. Vous devez désormais remplir cette colonne manuellement.

## 10.53 1.7.4 (2016-07-06)

#### Corrections de bugs

- Correction du script d'installation des tables liées au Contact flore (5a1fb07)
- Mise en cohérence avec GeoNature-mobile utilisant les classes 'gasteropodes' et 'bivalves' et non la classe générique 'mollusques'.

#### Nouveautés

- Corrections de mise en forme de la documentation
- Ajout de la liste rouge France de TaxRef lors d'une nouvelle installation (f4be2b6). A ne pas prendre en compte dans le cas d'une mise à jour.
- Ajout du MCD de la BDD - [https://github.com/PnX-SI/GeoNature/blob/master/docs/2016-04-29-mcd\\_geonaturedb.png](https://github.com/PnX-SI/GeoNature/blob/master/docs/2016-04-29-mcd_geonaturedb.png)

#### Note de version

- Vous pouvez passer directement de la version 1.6.0 à la 1.7.4 mais en vous référant aux notes de version de la 1.7.0.



- Remplacer `id_classe_mollusques` par `id_classe_gasteropodes` dans `web/js/config.js` et renseigner la valeur en cohérence avec l'`id_liste` retenu dans la table `taxonomie.bib_listes` pour les gastéropodes. Attention, vous devez avoir établi une correspondance entre les taxons gastéropodes et bivalves et leur liste dans la table `taxonomie.cor_taxon_liste`.

## 10.54 1.7.3 (2016-05-19)

### Corrections de bugs

- Correction de coordonnées vides dans l'export de Flore station. cf <https://github.com/PnX-SI/GeoNature/commit/0793a3d3d2b3719ed515058d1a0ba9baf7cb2096>
- Correction des triggers en base concernant un bug de saisie pour les taxons dont le taxon de référence n'est pas présent dans `taxonomie.bib_taxons`.

### Note de version

Rappel : commencez par suivre la procédure classique de mise à jour. <http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>

- Vous pouvez passer directement de la version 1.6.0 à la 1.7.3 mais en vous référant aux notes de version de la 1.7.0.
- Pour passer de la 1.7.2 à la 1.7.3 vous devez exécuter le script [https://github.com/PnX-SI/GeoNature/blob/master/data/update\\_1.7.2to1.7.3.sql](https://github.com/PnX-SI/GeoNature/blob/master/data/update_1.7.2to1.7.3.sql).

## 10.55 1.7.2 (2016-04-27)

### Corrections de bug

- Correction d'un bug dans l'export XLS depuis Flore Station.

### Note de version

- Vous pouvez passer directement de la version 1.6.0 à la 1.7.2 mais en vous référant aux notes de version de la 1.7.0.

## 10.56 1.7.1 (2016-04-27)

### Corrections de bug

- Ajout des listes flore manquantes dans le script de mise à jour `data/update_1.6to1.7.sql`.

## 10.57 1.7.0 (2016-04-24)

### Nouveautés

- Ajout du contact flore
- Correction et compléments dans les statistiques et mise en paramètre de leur affichage ou non, ainsi que de la date de début à prendre en compte pour leur affichage.

- Ajout d'un module d'export des données permettant d'offrir, en interne ou à des partenaires, un lien de téléchargement des données basé sur une ou des vues de la base de données (un fichier par vue). Voir <http://docs.geonature.fr>
- Modification des identifiants des listes pour compatibilité avec les applications GeoNature-Mobile.
- Complément dans la base de données pour compatibilité avec les applications GeoNature-Mobile.
- Correction d'une erreur sur l'importation de shape pour la recherche géographique
- WMS : correction de la liste des sites N2000, correction de l'affichage de l'aire optimale d'adhésion des parcs nationaux et retrait des sites inscrits et classés
- Correction d'un bug permettant la saisie d'une date d'observation postérieure à aujourd'hui dans Flore station
- Mention de la version de taxref sur la page d'accueil

### Note de version

Rappel : commencez par suivre la procédure classique de mise à jour. <http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>

#### 1. Modification des identifiants des listes de taxons pour compatibilité avec les applications GeoNature-Mobile.

Dans GeoNature-Mobile, les taxons sont filtrables par classe sur la base d'un `id_classe`. Ces id sont inscrits en dur dans le code des applications mobiles.

Dans la base GeoNature les classes taxonomiques sont configurables grace au vues `v_nomade_classes` qui utilisent les listes (`taxonomie.bib_listes`).

Les `id_liste` ont donc été mis à jour pour être compatibles avec les `id_classe` des applications mobiles.

Voir le script SQL d'update `data/update_1.6to1.7.sql` et LIRE ATTENTIVEMENT LES COMMENTAIRES.

- En lien avec les modifications ci-dessus, mettre à jour les variables des classes taxonomiques correspondant aux modification des `id_liste` dans `web/js/config.js`
- Ajouter dans le fichier `lib/sfGeonatureConfig.php` les variables `$struc_abregee`, `$struc_long`, `$taxref_version`, `$show_statistiques` et `$init_date_statistiques` (voir le fichier `lib/sfGeonatureConfig.php.sample`)

#### 2. Pour ajouter le Contact flore

- Exécuter le script sql `data/2154/contactflore.sql`
- Ajouter les variables `$id_lot_cflore = 7`, `$id_protocole_cflore = 7`, `$id_source_cflore = 7` et `$appname_cflore = 'Contact flore - GeoNature'`; dans `lib/sfGeonatureConfig.php` (voir le fichier d'exemple `lib/sfGeonatureConfig.php.sample`)
- Ajouter les variables `id_lot_contact_flore = 7`, `id_protocole_contact_flore = 7`, `id_source_contactflore = 7` dans `web/js/config.js` (voir le fichier d'exemple `web/js/config.js.sample`)
- l'enregistrement correspondant au contact flore dans la table `synthese.bib_sources` doit être actif (dernière colonne) pour que le contact flore soit accessible depuis la page d'accueil.

#### 3. Afin de mettre à jour la configuration WMS, vous devez exécuter le fichier `wms/update1.6to1.7.sh`.

Au préalable, assurez vous que les informations renseignées dans le fichier `config/settings.ini` sont à jour. L'ancien fichier sera sauvegardé sous `wms/wms_1.6.map`. Vous pourrez faire le choix de conserver ou de supprimer ce fichier de sauvegarde qui ne sera pas utilisé par l'application.

```
./wms/update1.6to1.7.sh
```

#### 4. Mise en place du module d'export

- Créer les vues retournant les données attendues.
- Configurer le module dans le fichier `lib/sfGeonatureConfig.php` à partir de l'exemple du fichier `lib/sfGeonatureConfig.php.sample`; section configuration du module d'export
  - Vous pouvez paramétrer plusieurs modules avec un nom pour chacun grace au paramètre `exportname`
  - Pour chacun des modules seuls les utilisateurs de geonature dont le `id_role` figure dans le tableau `authorized_roles_ids` peuvent exporter les données mises à disposition par le module d'export.
  - Chaque module peut comporter autant que vues que nécessaire (un bouton par vue générera un fichier zip par vue). Renseigner le tableau `views` pour chacun des modules.
  - Voir la documentation ici : <http://docs.geonature.fr>
- Attribution des droits nécessaires pour le répertoire permettant l'enregistrement temporaire des fichiers générés par le module d'export.

```
chmod -R 775 web/uploads/exports
```

- Rétablir les droits d'écriture et vider le cache

```
chmod -R 777 cache/
chmod -R 777 log/
php symfony cc
```

## 10.58 1.6.0 (2016-01-14)

### Note de version

- Pour les changements dans la base de données vous pouvez exécuter le fichier `data/update_1.5to1.6.sql`
- Mise à jour de la configuration Apache. Modifier le fichier `apache/wms.conf` en vous basant sur l'exemple <https://github.com/PnX-SI/GeoNature/blob/master/apache/wms.conf.sample#L16-L17>
- Ajouter le paramètre `$id_application` dans `lib/sfGeonatureConfig.php.php` (voir la valeur utilisée pour GeoNature dans les tables `utilisateurs.t_applications` et `utilisateurs.cor_role_droit_application`)
- Ajouter le paramètre `import_shp_projection` dans `web/js/configmap.map` - voir l'exemple dans le fichier <https://github.com/PnX-SI/GeoNature/blob/master/web/js/configmap.js.sample#L35>
- Supprimer toute référence à `gps_user_projection` dans `web/js/configmap.map`
- Ajouter un tableau JSON des projections disponibles pour l'outil de pointage GPS : `gps_user_projections` dans `web/js/configmap.map`. Respecter la structure définie dans <https://github.com/PnX-SI/GeoNature/blob/master/web/js/configmap.js.sample#L7-L14>. Attention de bien respecter la structure du tableau JSON et notamment sa syntaxe (accolades, virgules, nom des objects, etc...)
- Ajouter les `id_liste` pour les classes faune filtrables dans les formulaires de saisie dans le fichier `web/js/config.map`. Ceci concerne les variables `id_classe_oiseaux`, `id_classe_mammiferes`, `id_classe_amphibiens`, `id_classe_reptiles`, `id_classe_poissons` et

`id_classe_ecrevisses,` `id_classe_insectes,` `id_classe_arachnides,`  
`id_classe_myriapodes` et `id_classe_mollusques`. Voir l'exemple dans le fichier <https://github.com/PnX-SI/GeoNature/blob/master/web/js/config.js.sample#L32-44>

- Taxref a été mis à jour de la version 7 à 8. GeoNature 1.6.0 peut fonctionner avec la version 7. Cependant il est conseillé de passer en taxref V8 en mettant à jour la table `synthese.taxref` avec la version 8. Cette mise à jour pouvant avoir un impact fort sur vos données, son automatisation n'a pas été prévue. Le script SQL de migration de vos données de taxref V7 vers taxref V8 n'est donc pas fourni. Pour une installation nouvelle de la base de données, GeoNature 1.6.0 est fourni avec taxref V8.
- Le routing a été mis à jour, vous devez vider le cache de Symfony pour qu'il soit pris en compte. Pour cela, placez vous dans le répertoire racine de l'application et effectuez la commande suivante :

```
php symfony cc
```

### Changements

- Les recherches dans la synthèse sont désormais faites sur le `cd_ref` et non plus sur le `cd_nom` pour retourner tous les synonymes du taxon recherché - Fix #92
- Passage de taxref V7 à Taxref V8 - Fix #34
- Intégration de la première version de l'API permettant d'intégrer des données dans la synthèse depuis une source externe - [https://github.com/PnX-SI/GeoNature/blob/master/docs/geonature\\_webapi\\_doc.rst](https://github.com/PnX-SI/GeoNature/blob/master/docs/geonature_webapi_doc.rst)
- Mise en paramètre du `id_application` dans `lib/sfGeonatureConfig.php.php` - Fix #105
- Recharger la synthèse après suppression d'un enregistrement - Fix #94
- L'utilisateur peut lui-même définir le système de coordonnées dans l'outil de pointage GPS - Fix #107
- Mise en paramètre de la projection de la shape importée comme zone de recherche dans la synthèse
- Les exports XLS et SHP comportent le `cd_nom` ET le `cd_ref` de tous les synonymes du nom recherché ainsi que le `nom_latin` (`bib_taxons`) ET le `nom_valide` (`taxref`) - Fix #92
- SAISIE invertébrés - Ajout d'un filtre Mollusques - Fix #117
- Amélioration du vocabulaire utilisé sur la page d'accueil - #118
- Affichage d'un message pendant le chargement des exports
- Mise en place de statistiques automatiques sur la page d'accueil, basées sur les listes de taxons. A compléter.

### Corrections de bug

- Intégration de la librairie `OpenLayers.js` en local dans le code car les liens distants ne fonctionnaient plus - Fix #97
- Correction d'une erreur lors de l'enregistrement de la saisie invertébrés - Fix #104
- Correction d'une erreur de redirection si on choisit "Quitter" après la saisie de l'enregistrement (contact faune, mortalité et invertébrés) - Fix #102
- Correction du trigger `contactfaune.synthese_update_cor_role_fiche_cf()` - Fix #95
- Correction d'un bug dans les listes déroulantes des taxons filtrée par classe qui n'affichaient rien - Fix #109
- Correction d'un bug sur le contenu des exports shape avec le critère de protection activé - Fix #114
- Correction et adaptation faune-flore des exports shape
- SYNTHESE - Correction de la liste des taxons sans nom français - Fix #116
- Corrections CSS sur la page d'accueil - Fix #115
- Correction sur la largeur de la liste des résultats de la synthèse - Fix #110

- Correction des doublons dans la recherche multi-taxons - Fix #101
- Autres corrections mineures

## 10.59 1.5.0 (2015-11-26)

### Note de version

- Pour les changements dans la base de données vous pouvez exécuter le fichier `data/update_1.4to1.5.sql`
- Le bandeau de la page d'accueil `web/images/bandeau_faune.jpg` a été renommé en `bandeau_geonature.jpg`. Renommez le votre si vous aviez personnalisé ce bandeau.
- Si vous souhaitez désactiver certains programmes dans le “Comment ?” de la synthèse vous devez utiliser le champs actif de la table `meta.bib_programmes`.
- Compléter si nécessaire les champs `url`, `target`, `picto`, `groupe` et `actif` dans la table `synthese.bib_sources`.
- Nouvelle répartition des paramètres de configuration javascript en 2 fichiers (`config.js` et `configmap.js`). Vous devez reprendre vos paramètres de configuration du fichier `web/js/config.js` et les ventiler dans ces deux fichiers.
- Ajouter le paramètre `id_source_mortalite = 2;` au fichier `web/js/config.js`;
- Retirer le paramètre `fuseauUTM;` du fichier `web/js/config.js`;
- Bien définir le système de coordonnées à utiliser pour les pointages par coordonnées fournies en renseignant le paramètre `gps_user_projection` dans le fichier `web/js/configmap.js`;
- Ajouter le paramètre `public static $id_source_mortalite = 2;` au fichier `lib/sfGeonatureConfig.php`;
- Ajouter le paramètre `public static $srid_ol_map = 3857;` au fichier `lib/sfGeonatureConfig.php`;
- L'altitude est calculée automatiquement à partir du service “Alticodage” de l'API GeoPortail de l'IGN et non plus à partir de la couche `layers.l_isolines20`. Ajoutez ce service dans votre contrat API Geoportail. Il n'est donc plus nécessaire de remplir la couche `layers.l_isolines20`. Cette couche peut toutefois encore être utile si l'utilisateur supprime l'altitude calculée par l'API Geoportail dans les formulaires de saisie.
- Le loup et le lynx sont retirés par défaut de la saisie (saisie recommandée dans le protocole national du réseau grands prédateurs)
- Le cerf, chamois et le bouquetin doivent être saisis selon 6 critères de sexe et age et non 5 comme les autres taxons. Comportement peut-être changé en modifiant la vue `contactfaune.v_nomade_taxons_faune`.
- Mortalité est désormais une source à part entière alors qu'elles étaient mélangées avec la source ContactFaune précédemment. Si vous avez déjà des données de mortalité enregistrées, vous devez adapter la requête SQL ci-dessous avec votre `id_source` pour Mortalité et l'exécuter :

```
UPDATE synthese.syntheseeff SET id_source = 2 WHERE id_source = 1 AND id_
↳critere_synthese = 2;
```

### Changements

- Optimisation des vues aux chargement des listes de taxons. Fixes #64
- Généricité des champs dans `meta.bib_programmes` (champs `sitpn` renommé en `public`). Fixes #68

- Ajout d'un champ `actif` à la table `meta.bib_programmes` permettant de masquer certains programmes dans le "Comment ?" de la synthèse. Fixes #66
- Ajout d'un champ `url`, `target`, `picto`, `groupe` et `actif` dans la table `synthese.bib_sources` pour générer la page d'accueil dynamiquement et de manière générique. Fixes #69
- Construire dynamiquement la liste des liens vers la saisie des différents protocoles à partir de la table `synthese.bib_sources`. Fixes #69
- Tous les styles des éléments de la page d'accueil ont été passés en CSS. Fixes #57
- Amélioration de l'interface pendant le chargement des différentes applications (synthèse, flore station, formulaires de saisie...). Fixes #65
- Recentrage sur la position de l'utilisation en utilisant le protocole de géolocalisation intégré au navigateur de l'utilisateur. Fixes #65
- Un message automatique conseille les utilisateurs d'Internet Explorer de plutôt utiliser Firefox ou Chrome. Fixes #65
- Tri par défaut par date décroissante des 50 dernières observations affichées à l'ouverture de la Synthèse. Fixes #51
- Vocabulaire. "Dessiner un point" remplacé par "Localiser l'observation". Fixes #66
- Mise à jour des copyrights dans les pieds de page de toutes les applications.
- Refonte du CSS du formulaire de login avec bootstrap et une image de fond différente.
- Refonte Bootstrap de la page d'accueil.
- Homogénéisation du pied de page.
- FloreStation et Bryophytes - Homogénéiser interaction carte liste - ajout d'un popup au survol. Fixes #74
- Suppression d'images non utilisées dans le répertoire `web/images`.
- Mise en cohérence des vues taxonomiques faune. Fixes #81
- Calcul de l'altitude à partir du service "Alticodage" de l'API GeoPortail de l'IGN.
- Factorisation et généralisation du module permettant un positionnement des pointages par saisie de coordonnées selon projection et bbox fournies en paramètres de config.
- Création d'une configuration javascript carto dédiée (`configmap.js`).

#### Corrections de bug

- Correction des problèmes de saisie de la version 1.4.0 liés à la migration de la taxonomie.
- Correction de bugs dans Flore Station et Bryophytes (Zoom, recherche)

## 10.60 1.4.0 (2015-10-16)

#### Note de version

- La gestion de la taxonomie a été mis en conformité avec le schéma `taxonomie` de la base de données de TaxHub (<https://github.com/PnX-SI/TaxHub>). Ainsi le schéma `taxonomie` intégré à GeoNature 1.3.0 doit être globalement revu. L'ensemble des modifications peuvent être réalisées en exécutant la partie correspondante dans le fichier `data/update_1.3to1.4.sql` ([https://github.com/PnX-SI/GeoNature/blob/master/data/update\\_1.3to1.4.sql](https://github.com/PnX-SI/GeoNature/blob/master/data/update_1.3to1.4.sql)).
- De nouveaux paramètres ont potentiellement été ajoutés à l'application. Après avoir récupéré le fichier de configuration de votre version 1.3.0, vérifiez les changements éventuels des différents fichiers de configuration.

- Modification du nom de l'host hébergeant la base de données. databases -> geonatdbhost. A changer ou ajouter dans le `/etc/hosts` si vous avez déjà installé GeoNature.
- Suivez la procédure de mise à jour : <http://docs.geonature.fr/installation-standalone.html#mise-a-jour-de-l-application>

### Changements

- A l'installation initiale, chargement en base des zones à statuts juridiques pour toute la France métropolitaine à partir des sources de l'INPN
- A l'installation initiale, chargement en base de toutes les communes de France
- Mise en place de la compatibilité de la base avec le schema de TaxHub

## 10.61 1.3.0 (2015-02-11)

Pré-Version de GeoNature - Faune ET Flore. Le fonctionnement de l'ensemble n'a pas été totalement testé, des bugs sont identifiés, d'autres subsistent certainement.

### Changements

- Grosse évolution de la base de données
- Ajout de deux applications de saisie flore (flore station et bryophytes)
- Intégration de la flore en sythese
- Ajouter un `id_lot`, `id_organisme`, `id_protocole` dans toutes les tables pour que ces id soit ajoutés vers la sythese en trigger depuis les tables et pas avec des valeurs en dur dans les triggers. Ceci permet d'utiliser les paramètres de conf de GeoNature
- Ajout d'une fonction à la base pour correction du dysfonctionnement du wms avec mapserver
- Suppression du champ `id_taxon` en sythese et lien direct de la sythese avecle taxref. ceci permet d'ajouter des données en sythese directement dans la base sans ajouter tous les taxons manquants dans la table `bib_taxons`
- Suppression de la notion de coeur dans les critère de recherche en sythese
- Ajout d'un filtre faune flore fonge dans la sythese
- Ajout de l'embranchement et du regne dans les exports
- Permettre à des partenaires de saisir mais d'exporter uniquement leurs données perso
- Ajout du déterminateur dans les formulaires invertébrés et contactfaune + en sythese
- Ajout du référentiel géographique de toutes les communes de France métropolitaine
- Ajout des zones à statuts juridiques de la région sud-est (national à venir)
- Bugs fix

### BUG à identifier

Installation :

- corriger l'insertion de données flore station qui ne fonctionne pas

Bryophytes :

- Corriger la recherche avancée par date sans années

Synthèse :

- la construction de l'arbre pour choisir plusieurs taxons ne tient pas compte des filtres

- le fonctionnement des unités géographiques n'a pas été testé (initialement conçu uniquement pour la faune)

## 10.62 1.2.0 (2015-02-11)

Version stabilisée de GeoNature - Faune uniquement (Synthèse Faune + Saisie ContactFauneVertebre, ContactFauneInvertebre et Mortalité).

### Changements

- Modification du nom de l'application de FF-synthese en GeoNature
- Changement du nom des utilisateurs PostgreSQL
- Changement du nom de la base de données
- Mise à jour de la documentation (<http://docs.geonature.fr>)
- Automatisation de l'installation de la BDD
- Renommer les tables pour plus de généralité
- Supprimer les tables inutiles ou trop spécifiques
- Gestion des utilisateurs externalisée et centralisée avec UsersHub (<https://github.com/PnX-SI/UsersHub>)
- Correction de bugs
- Préparation de l'intégration de la Flore pour passer de GeoNature Faune à GeoNature Faune-Flore

## 10.63 1.1.0 (2014-12-11)

### Changements

- Modification du schéma de la base pour être compatible taxref v7
- Import automatisé de taxref v7
- Suppression des tables de hiérarchie taxonomique (famille, ordre, ...) afin de simplifier l'utilisation de la taxonomie.
- Création de la notion de groupe (para-taxonomique) à la place de l'utilisation des classes.
- Ajout de données pour pouvoir tester de façon complète l'application (invertébrés, vertébrés)
- Ajout de données exemples
- Bugs fix

## 10.64 1.0.0 (2014-12-10)

Version fonctionnelle des applications : visualisation de la synthèse faune, saisie d'une donnée de contact (vertébrés, invertébrés, mortalité)

### Changements

- Documentation de l'installation d'un serveur Debian wheezy pas à pas
- Documentation de la mise en place de la base de données
- Documentation de la mise en place de l'application et de son paramétrage



- Script d'insertion d'un jeu de données test
- Passage à PostGIS v2
- Mise en paramètre de la notion de lot, protocole et source

#### **Prochaines évolutions**

- Script d'import de taxref v7
- Utilisation préférentielle de la taxonomie de taxref plutôt que les tables de hiérarchie taxonomique

## **10.65 0.1.0 (2014-12-01)**

- Création du projet et de la documentation



## HTTP ROUTING TABLE

<b>/admin</b>	<b>??</b>
GET /admin/bibnomenclaturestypesadmin/, ??	POST /admin/tnomenclaturesadmin/delete/, ??
GET /admin/bibnomenclaturestypesadmin/details/, ??	POST /admin/tnomenclaturesadmin/edit/, ??
GET /admin/bibnomenclaturestypesadmin/edit/, ??	POST /admin/tnomenclaturesadmin/new/, ??
GET /admin/bibnomenclaturestypesadmin/new/, ??	
	<b>/config</b>
GET /admin/static/ (path:filename), ??	GET /config, ??
GET /admin/tadditionalfields/, ??	
GET /admin/tadditionalfields/details/, ??	<b>/geo</b>
GET /admin/tadditionalfields/edit/, ??	GET /geo/areas, ??
GET /admin/tadditionalfields/new/, ??	GET /geo/municipalities, ??
GET /admin/tnomenclaturesadmin/, ??	POST /geo/altitude, ??
GET /admin/tnomenclaturesadmin/details/, ??	POST /geo/area_size, ??
GET /admin/tnomenclaturesadmin/edit/, ??	POST /geo/areas, ??
	POST /geo/info, ??
	<b>/gn_auth</b>
GET /admin/tnomenclaturesadmin/new/, ??	GET /gn_auth/login_cas, ??
POST /admin/bibnomenclaturestypesadmin/action/, ??	GET /gn_auth/logout_cruved, ??
POST /admin/bibnomenclaturestypesadmin/ajax/update/, ??	POST /gn_auth/login_cas, ??
	<b>/gn_commons</b>
POST /admin/bibnomenclaturestypesadmin/delete/, ??	GET /gn_commons/list/parameters, ??
POST /admin/bibnomenclaturestypesadmin/edit/, ??	GET /gn_commons/media/ (int:id_media), ??
POST /admin/bibnomenclaturestypesadmin/new/, ??	GET /gn_commons/media/thumbnails/ (int:id_media) / (int:id_media), ??
POST /admin/tadditionalfields/action/, ??	GET /gn_commons/m medias/ (string:uuid_attached_row), ??
POST /admin/tadditionalfields/ajax/update/, ??	GET /gn_commons/modules, ??
POST /admin/tadditionalfields/delete/, ??	GET /gn_commons/t_mobile_apps, ??
POST /admin/tadditionalfields/edit/, ??	POST /gn_commons/media, ??
POST /admin/tadditionalfields/new/, ??	POST /gn_commons/media/ (int:id_media), ??
POST /admin/tnomenclaturesadmin/action/, ??	PUT /gn_commons/media, ??
	PUT /gn_commons/media/ (int:id_media), ??
POST /admin/tnomenclaturesadmin/ajax/update/, ??	DELETE /gn_commons/media/ (int:id_media), ??

```

/gn_monitoring                                PATCH /meta/dataset/(int:id_dataset),
GET /gn_monitoring/siteareas/(int:id_site),    ??
??
GET /gn_monitoring/siteslist,??                /monitorings
GET /gn_monitoring/siteslist/(int:id_site),    GET /monitorings/config,??
??                                              GET /monitorings/config/(string:module_code),
??                                              ??
/gn_profiles                                GET /monitorings/exports/csv/(module_code)/(type)/
GET /gn_profiles/consistancy_data/(id_synthese),??
??                                              GET /monitorings/module/(value),??
GET /gn_profiles/cor_taxon_phenology/(int:cd_ref),    GET /monitorings/modules,??
??                                              GET /monitorings/object/(string:module_code)/(string:
GET /gn_profiles/valid_profile/(int:cd_ref),    ??
??                                              GET /monitorings/object/(string:module_code)/(string:
POST /gn_profiles/check_observation,??          ??
GET /monitorings/object/module,??
/habref                                     GET /monitorings/routes_list,??
GET /habref/correspondance/(int:cd_hab),    GET /monitorings/test_permission,??
??                                              GET /monitorings/util/(string:type_util)/(int:id),
??                                              ??
GET /habref/habitat/(int:cd_hab),??          GET /monitorings/util/(string:type_util)/(string:
GET /habref/habitats/autocomplete,??          ??
GET /habref/search/(field)/(ilike),??        GET /monitorings/util/init_data/(string:module_code)
GET /habref/typo,??                          ??
GET /monitorings/util/nomenclature/(string:code_nor
/meta                                     ??
GET /meta/acquisition_framework/(id_acquisition_framework),
??                                            POST /monitorings/exports/pdf/(module_code)/(object
??                                            ??
GET /meta/acquisition_framework/(id_acquisition_framework)/bbox,
??
/nomenclatures/nomenclature
GET /meta/acquisition_framework/(id_acquisition_framework)/stats,
??                                            GET /nomenclatures/nomenclature/(int:id_type),
??                                            ??
GET /meta/acquisition_framework/publish/(int:af_id),
??                                            GET /nomenclatures/nomenclature/(string:code_type),
??                                            ??
GET /meta/acquisition_frameworks,??
GET /meta/acquisition_frameworks/export_pdf/(id_acquisition_framework),
??
/nomenclatures/nomenclatures
GET /meta/dataset/(int:id_dataset),??        GET /nomenclatures/nomenclatures,??
GET /meta/dataset/export_pdf/(id_dataset),??
??
/occhab
GET /meta/datasets,??                        GET /occhab/defaultNomenclatures,??
GET /meta/list/acquisition_frameworks,    GET /occhab/station/(int:id_station),
??                                          ??
GET /meta/sensi_report,??                  GET /occhab/stations,??
GET /meta/uuid_report,??                  POST /occhab/export_stations/(export_format),
??                                          ??
POST /meta/acquisition_framework,??        POST /occhab/station,??
POST /meta/acquisition_framework/(int:id_acquisition_framework),
??                                          DELETE /occhab/station/(int:id_station),
??                                          ??
POST /meta/dataset,??
POST /meta/dataset/(int:id_dataset),??
POST /meta/upload_canvas,??
DELETE /meta/acquisition_framework/(int:af_id),
??
/occtax
DELETE /meta/dataset/(int:ds_id),??          GET /occtax/counting/(int:id_counting),
??                                          ??
GET /occtax/defaultNomenclatures,??
GET /occtax/export,??

```

GET /occtax/occurrences,??	POST /permissions_backoffice/other_permissions_form,??
GET /occtax/releve/(int:id_releve),??	??
GET /occtax/releves,??	POST /permissions_backoffice/other_permissions_form,??
GET /occtax/vreleveocctax,??	??
POST /occtax/occurrence/(int:id_occurrence),??	/pypn
POST /occtax/only/releve,??	GET /pypn/register/test_uh,??
POST /occtax/only/releve/(int:id_releve),??	POST /pypn/register/post_usershub/(string:type_act),??
POST /occtax/releve,??	
POST /occtax/releve/(int:id_releve)/occurrence,??	/synthese
DELETE /occtax/occurrence/(int:id_occ),??	GET /synthese,??
DELETE /occtax/releve/(int:id_releve),??	GET /synthese/color_taxon,??
DELETE /occtax/releve/occurrence_counting/(int:id_count),??	GET /synthese/defaultsNomenclatures,??
	GET /synthese/export_metadata,??
	GET /synthese/for_web,??
	GET /synthese/general_stats,??
	GET /synthese/observation_count,??
	GET /synthese/observation_count_per_column/(column),??
/permissions	
GET /permissions/cruved,??	GET /synthese/observations_bbox,??
GET /permissions/logout_cruved,??	GET /synthese/sources,??
/permissions_backoffice	GET /synthese/taxa_count,??
GET /permissions_backoffice/cruved_form/module_synthese/taxons_tree,??	GET /synthese/taxa_distribution,??
GET /permissions_backoffice/cruved_form/module_synthese/taxons_tree,??	GET /synthese/taxons_tree,??
GET /permissions_backoffice/filter_form/module_synthese/export_metadata,??	POST /synthese/export_metadata,??
GET /permissions_backoffice/filter_form/module_synthese/export_metadata,??	POST /synthese/export_observations,??
GET /permissions_backoffice/filter_form/module_synthese/export_metadata,??	POST /synthese/export_observations,??
GET /permissions_backoffice/filter_form/module_synthese/export_metadata,??	POST /synthese/export_taxons,??
GET /permissions_backoffice/filter_list/module_synthese/for_web,??	POST /synthese/for_web,??
GET /permissions_backoffice/other_permissions/module_synthese/reports,??	POST /synthese/reports,??
GET /permissions_backoffice/other_permissions/module_synthese/reports,??	PUT /synthese/reports/(int:id_report),??
GET /permissions_backoffice/other_permissions_form/user/(int:id_role)/filter_type/(int:id_filter_type),??	/users
GET /permissions_backoffice/user_cruved/module_synthese/confirmation,??	GET /users/confirmation,??
GET /permissions_backoffice/user_other_permissions/module_synthese/menu,??	GET /users/menu/(int:id_menu),??
GET /permissions_backoffice/users,??	GET /users/organisms,??
POST /permissions_backoffice/cruved_form/module_synthese/role/confirmation,??	GET /users/role/(int:id_role),??
POST /permissions_backoffice/cruved_form/module_synthese/role/confirmation,??	POST /users/role/(int:id_role)/object/(int:id_object),??
POST /permissions_backoffice/filter/(int:id_filter_type)/id_filter/(int:id_filter_type),??	POST /users/inscription,??
POST /permissions_backoffice/filter_form/module_synthese/login/recovery,??	POST /users/login/recovery,??
POST /permissions_backoffice/filter_form/module_synthese/login/recovery,??	PUT /users/password/change,??
POST /permissions_backoffice/filter_form/module_synthese/login/recovery,??	PUT /users/password/change,??
POST /permissions_backoffice/filter_form/module_synthese/login/recovery,??	PUT /users/role,??
POST /permissions_backoffice/filter_form/module_synthese/login/recovery,??	/validation
	GET /validation,??

GET /validation/date/(uuid:uuid),??  
POST /validation,??