

Rapport de stage

Master 2 - Géographies Numériques

Promotion 2021/2022

**GÉO
NUM**

Auteure : Cendrine HOARAU

Tuteur professionnel : Jean-Pascal MILCENT

Tuteur académique : Thierry JOLIVEAU



MISE A JOUR ET DEVELOPPEMENT GEONATURE AU CBNA

Stage effectué entre mars et septembre 2022



Panicaud des Alpes (crédits : CBNA)

SOMMAIRE

Remerciements	3
Glossaire	3
Introduction.....	4
CONTEXTE.....	4
LE CBNA	5
Les Conservatoires Botaniques Nationaux.....	5
Le Conservatoire Botanique National Alpin	6
Le SI du CBNA	7
GEONATURE	12
L'environnement applicatif	12
GeoNature comme outil métier au CBNA	14
LES OBJECTIFS DU STAGE	15
AMÉLIORATION DE GEONATURE EXPERT	17
MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL.....	17
AMÉLIORER LA VISUALISATION DES DONNÉES GÉOGRAPHIQUES	18
Agrégation des données par géométrie.....	19
Affichage des données par mailles.....	23
INTÉGRER LES DÉVELOPPEMENTS.....	31
Fusionner les branches de travail.....	31
Les difficultés rencontrées avec une application collaborative	32
MIGRATION DES MODULES CONSERVATION DE LA FLORE	33
MONTÉE EN VERSION DES MODULES	36
Packaging des modules	37
Création des révisions Alembic	37
MIGRATION VERS LE NOUVEAU SERVEUR	39
BILAN	40
Résultats et pistes d'amélioration.....	40
Montée en compétences	40
Perspectives	41
Bibliographie/Webographie	42
Index des figures	42
Annexes	44

Remerciements

En premier lieu, je tiens à remercier vivement l'encadrant de ce stage, Jean-Pasal Milcent, pour son accueil, sa bienveillance et sa confiance. Le partage de son expertise et de ses connaissances ainsi que son investissement m'ont été d'une aide très précieuse.

Merci aussi à l'ensemble du personnel du CBNA, et particulièrement Bertrand Lienard, directeur, pour m'avoir accueilli au sein de cette équipe dynamique et bienveillante.

Un grand merci à Nicolas Rouyer, avec qui j'ai partagé ces six mois de stage, qui sont passés d'autant plus vite lorsque nous avons travaillé ensemble.

Je remercie également les développeurs du pôle SI du PNE, Théo Lechémia et Elie Bouttier pour leur disponibilité et leur apport d'expertise technique tout au long du stage.

Je tiens également à remercier l'équipe encadrante du master pour leur disponibilité et leur suivi tout au long de l'année.

Glossaire

CBNA :	Conservatoire Botanique National Alpin
PNE :	Parc National des Ecrins
SI :	Système d'Information
SINP :	Système d'Information de l'INventaire du Patrimoine naturel
PACA :	Provence-Alpes-Côte d'Azur
AURA :	Auvergne-Rhône-Alpes
Silene :	Plateforme régionale du SINP PACA
Biodiv'AURA :	Plateforme régionale du SINP AURA
GN :	GeoNature
TH :	TaxHub
UH :	UsersHub
GH :	Github
BDD :	Base de données

Introduction

Ce stage de dernière année de Master "Géographies Numériques" s'inscrit dans la continuité de mes apprentissages académiques et professionnels, dont mon stage de première année de Master au sein du service SI du Parc national des Ecrins (PNE). C'est lors de ce stage que j'ai développé mon appétence pour les outils open source et collaboratifs. Cette immersion avec les développeurs d'environnements applicatifs d'envergure tels que GeoNature¹ et Geotrek² est à l'origine mon choix de stage de fin d'études. Les cours de développement web de M2 ont par la suite conforté ma volonté de monter en compétences en informatique, dans l'objectif de pouvoir contribuer au développement d'outils cartographiques libres, utilisés par une grande communauté d'acteurs.

CONTEXTE

Dans le cadre du réseau **Flore Sentinelle**³ animé par le Conservatoire Botanique National Alpin (CBNA), plus de 25 structures dédiées à la connaissance et à la conservation de la flore sur l'arc alpin français se sont regroupées pour développer des protocoles communs de suivis des espèces et des habitats patrimoniaux.

Des outils de saisie, de stockage et de gestion de ces données, accessible à tous les partenaires ont été développés en s'appuyant sur l'outil **GeoNature** et son écosystème applicatif. GeoNature est une application web pour la gestion et la diffusion des données faune et flore. C'est un outil open-source⁴ reconnu dans les milieux des systèmes d'informations naturalistes qui est aujourd'hui utilisé par plus d'une soixantaine de structures en France.

Dès 2016, afin de faciliter le travail de suivi des dynamiques démographiques des espèces végétales et de leurs habitats, deux modules GeoNature ont été développés : '**Suivi Flore Territoire**'⁵ et '**Suivi Habitat Territoire**'⁶. Le développement de ces modules est piloté par le CBNA et s'inscrit dans le projet Flore Sentinelle.

Dans le cadre du projet **FloreClim** qui vise à évaluer l'impact du changement climatique sur la flore alpine, deux autres modules GeoNature ont été développés pour optimiser les stratégies de conservation et de gestion de la flore : '**Flore Prioritaire**'⁷ et '**Suivi Habitat Station**'⁸.

¹ <https://geonature.fr/>

² <https://geotrek.fr/>

³ <https://floresentinelle.fr/>

⁴ <https://github.com/PnX-SI/GeoNature>

⁵ <https://github.com/PnX-SI/GeoNature>

⁶ https://github.com/PnX-SI/gn_module_suivi_habitat_territoire

⁷ https://github.com/PnX-SI/gn_module_flore_prioritaire

⁸ https://github.com/PnX-SI/gn_module_suivi_habitat_station

L'ensemble applicatif GeoNature ainsi que les quatre modules spécifiques à la conservation de la flore ont été déployés sur un serveur Linux sous Debian. Les besoins du réseau grandissant, le CBNA souhaite mettre à jour GeoNature vers une version plus récente, offrant plus de fonctionnalités.

En parallèle des projets Flore Sentinelle et FloreClim, le CBNA s'implique largement dans la mise en œuvre et la coordination des deux déclinaisons régionales du **SINP** (Système d'Information de l'INventaire du Patrimoine naturel) sur son territoire d'agrément : **Biodiv'AURA** en Auvergne-Rhône-Alpes et **Silene** en Provence-Alpes-Côte d'Azur. Les SINP ont été mis en place pour la collecte et la validation homogène des données scientifiques sur l'ensemble du territoire. Ils reposent sur un large réseau d'acteurs associant la recherche, les naturalistes, les gestionnaires d'espaces naturels, l'administration et les collectivités.

Depuis 2021, le CBNA a déployé les deux plateformes de diffusion de données des SINP AURA et PACA. Ces plateformes diffusent les données relatives à la flore, la fonge et la faune (vertébrée et invertébrée). Elles se déclinent chacune en deux outils : un Atlas⁹, outil de diffusion au grand public, et un outil Expert, destiné aux professionnels, permettant de visualiser et télécharger les données au format standard¹⁰ du SINP.

Récemment, les administrateurs du SINP PACA ont mandaté le CBNA pour améliorer la visualisation des données sur l'outil Silene Expert. Jusqu'à présent il n'est possible de voir les observations naturalistes sur la **Synthèse** de GeoNature que sous forme de points clusterisés (voir figure 8). La Synthèse est un outil central de Silene Expert puisqu'il permet de visualiser et de filtrer toutes les données d'une base GeoNature sur une carte Leaflet. Afin d'améliorer la visualisation des données à petite échelle, il a été demandé que l'on puisse également visualiser les données sous forme de mailles colorées en fonction du nombre d'observations qu'elles contiennent.

Ce stage s'inscrit suite aux besoins techniques grandissants sur GeoNature au CBNA.

LE CBNA

Les Conservatoires Botaniques Nationaux

Les Conservatoires botaniques nationaux sont des organismes agréés par le Ministère de la Transition écologique et solidaire. Le réseau des CBN regroupe 10 Conservatoires¹¹ botaniques nationaux en métropole et deux en outre-mer.

Les Conservatoires botaniques participent à l'inventaire du patrimoine naturel végétal et procèdent à l'identification et à la conservation des éléments rares et menacés de la flore, de la végétation et des habitats naturels et semi-naturels. Ils prêtent leur concours scientifique et

⁹ <https://atlas.biodiversite-auvergne-rhone-alpes.fr/> & <https://nature.Silene.eu/>

¹⁰ <https://standards-sinp.mnhn.fr/>

¹¹ http://cbnfc-ori.org/sites/cbnfc-ori.org/files/carte-reseau-cbn_2022_web.jpg

technique à l'État, aux établissements publics, ainsi qu'aux collectivités territoriales. Ils assurent une mission d'information et de sensibilisation du public aux enjeux liés à la flore et l'animation d'un réseau de collaborateurs (bénévoles, partenaires...).

Les CBN sont regroupés à l'intérieur d'une association régie par la loi du 1er juillet 1901. La Fédération des Conservatoires botaniques nationaux (FCBN)¹² assure une mission de représentation commune du réseau des CBN auprès des pouvoirs publics. La mission de coordination technique des CBN est assurée par l'Office Français de la Biodiversité (OFB).

Le Conservatoire Botanique National Alpin

Le Conservatoire Botanique National Alpin (CBNA) est un organisme public dédié à la connaissance et la préservation de la flore et des végétations des Alpes françaises et de leurs piémonts. Il est agréé par le ministère chargé de l'environnement et fait partie de la Fédération nationale des conservatoires botaniques nationaux.

L'équipe du Conservatoire botanique national alpin compte 35 personnes réparties entre le siège à Gap (05), l'antenne rhônalpine à Chambéry (73) et un bureau à Annecy (74). Elle est structurée en quatre services : Connaissance, Conservation, Systèmes d'information et un dernier service Administration, Finance et Communication. Elle comprend principalement des experts en botanique, écologie végétale et géomatique.

En tant que Conservatoire botanique national, le CBNA exerce cinq missions définies par le Code de l'environnement et précisées dans son agrément :



Figure 1 : Territoire d'agrément du CBNA, source: CBNA

- Le développement de la connaissance sur la flore, la fonge, les végétations et les habitats, aux échelles territoriales, nationale et biogéographiques.
- La gestion, diffusion et valorisation de données sur la flore, la fonge, les végétations et les habitats
- La contribution à la gestion conservatoire de la flore, de la fonge, des ressources phytogénétiques sauvages, des végétations, des habitats et des espaces, et à la restauration écologique
- L'appui à l'élaboration et à la mise en œuvre des politiques publiques et de la réglementation aux échelles territoriales, nationale et européenne
- La communication, sensibilisation et mobilisation des acteurs.

¹² <https://www.fcfn.fr/>

Son territoire d'intervention est centré sur la zone alpine française. Il est défini selon des critères biogéographiques et institutionnels. Il couvre les départements de l'Ain, des Alpes-de-Haute-Provence, des Hautes-Alpes, de la Drôme, de l'Isère, de la Savoie et de la Haute-Savoie.

Organisé sous la forme d'un syndicat mixte, le CBNA s'appuie sur un comité syndical qui guide ses choix d'actions à mettre en œuvre. Ce comité syndical est composé de la Ville de Gap, du Conseil départemental des Hautes-Alpes et des Régions Sud-Provence-Alpes-Côte d'Azur et Auvergne-Rhône-Alpes. Les Directions Régionales de l'Environnement de l'Aménagement et du Logement (DREAL) Auvergne-Rhône-Alpes et Provence-Alpes-Côte d'Azur sont des invitées permanentes au comité syndical du CBNA.

Depuis 25 ans le CBNA a construit un ensemble de ressources diversifiées sur la connaissance de la flore et des milieux alpins, leur distribution géographique et leur évolution:

- Des bases de données dont certaines ont leurs données exportées sur Biodiv'AURA et Silene Nature
- Une bibliothèque¹³ de documents et de photos
- Ses collections vivantes : banques séminales et culturelles
- Des herbiers
- Un jardin botanique alpin

Le SI du CBNA

• Rôle et fonctionnement

Le service SI occupe une position transversale puisqu'il est amené à travailler avec tous les services du CBNA. Il assiste aussi bien le service Conservation en mettant en place des modules de suivi floristique GeoNature, que le service Connaissance pour le stockage des données, ou encore le service Finance et Communication dans le maintien des outils de gestion de projets.

De par ses missions de préservation de la flore alpine, le CBNA est amené à collecter des quantités importantes de données spatialisées. Le rôle du SI est donc d'organiser le stockage des données et de faciliter la collecte et l'analyse de ces données via des outils métier adaptés. Les deux grandes composantes métier du SI tiennent donc dans l'administration de serveurs de base de données et dans le développement de solutions métier. Cependant, actuellement le service SI est davantage focalisé sur la gestion des données (import, export, transformation) et le maintien des solutions de stockage (administration serveur, développement).

¹³ <http://www.cbn-alpin-biblio.fr/>

Au CBNA, le service Système d'Informations est constitué de 6 personnes :

- Un chef de service et administrateur de données,
- Un informaticien généraliste (maintenance, réseau, téléphonie)
- Un géomaticien
- Un développeur
- Une technicienne SIG
- Une opératrice de saisie

Le SI du CBNA tend depuis quelques années à utiliser des outils informatiques libres comme Linux, QGIS et PostgreSQL. Cette décision rompt avec une logique d'enfermement liée aux logiciels propriétaires utilisés précédemment qui pouvaient poser des soucis de gouvernance et qui peuvent s'avérer onéreux.

Depuis 2020, le CBNA a fait le choix de se tourner vers GeoNature comme solution de gestion et de diffusion des données. Ce choix participe à la volonté de travailler avec des outils web libres, de structurer et centraliser les données métier ainsi que de mieux les valoriser sur des plateformes de diffusion. L'objectif étant d'homogénéiser la chaîne de travail du CBNA.

La saisie des données sur le terrain est, quant à elle, assez hétérogène. Elle se fait encore "manuellement" pour la plupart des agents, c'est-à-dire avec un GPS pour pointer ou créer des zones sur le terrain et un carnet de notes papier. Les données historiques sont saisies et stockées dans des fichiers Excel et sont importées dans les outils métiers via des scripts Bash ou manuellement par des opératrices de saisie.

L'objectif à terme est de concevoir des modules GeoNature qui reposent sur les protocoles mis en place par les services Conservation et Connaissance et qui puissent servir d'outil de saisie, de gestion et de diffusion des données. Cela permettrait d'éviter la dispersion méthodologique, notamment liée aux notes papier et aux fichiers Excel, de fluidifier le flux de travail ainsi que de minimiser la perte des données.

- **Le stockage des données**

Le système d'information du CBNA est constitué de bases de données thématiques.

Base documentaire

La base documentaire, utilisant le logiciel Kentika, permet la gestion du fonds documentaire du CBNA. L'ensemble de cette base de données est librement consultable en ligne, avec la possibilité, pour une partie des documents, en plus de l'accès à la notice bibliographique, d'accéder à la version numérique.

Base iconographique

La base iconographique permet la gestion du fonds iconographique du CBNA via le logiciel Ajaris. Une sélection de cette base de données est accessible au grand public sous licence libre ; le personnel du CBNA a accès à l'ensemble des photographies.

Bases Flore Sentinelle

Les bases de données Flore Sentinelle ont été développées en partenariat avec le Parc national des Écrins sur l'écosystème applicatif de GeoNature.

Base Flore-Habitats

Développée par le CBN Méditerranéen, cette base de données est utilisée par les CBN Méditerranéen, Corse et Alpin dans le cadre du projet de mutualisation des systèmes d'information. Les données sont accessibles sur l'application web SImethis, outil métier du CBNA, qui permet la gestion des informations sur la flore et les végétations issues de différentes sources (bibliographie, terrain, herbier...). Les observations floristiques issues de cette base sont diffusées dans les SINP régionaux et nationaux.

Base Conservation Ex-situ

Mise en place en 2014 sous Windev, à partir de fichiers indépendants préexistants, cette base de données permet la gestion des informations relatives aux banques gérées par le service Conservation :

- Banque de récolte
- Banque de culture
- Banque de semis
- Banque de semences
- Banque tests de germination

Cartographie des habitats

Ces informations issues de différentes sources (CBNA et partenaires) ne sont actuellement pas stockées dans une base de données, mais sous forme de fichiers SIG (shape) indépendants consultables en interne.

Données SIG de référence

Dans le cadre de son activité, le CBNA a besoin de données géographiques de référence (périmètres administratifs, espaces protégés, imagerie...) fournies par les services de l'Etat, l'IGN et le BRGM. Ces données existent sous forme de fichiers géographiques indépendants et sont stockées en base de données spatiale et sont accessibles au personnel du CBNA.

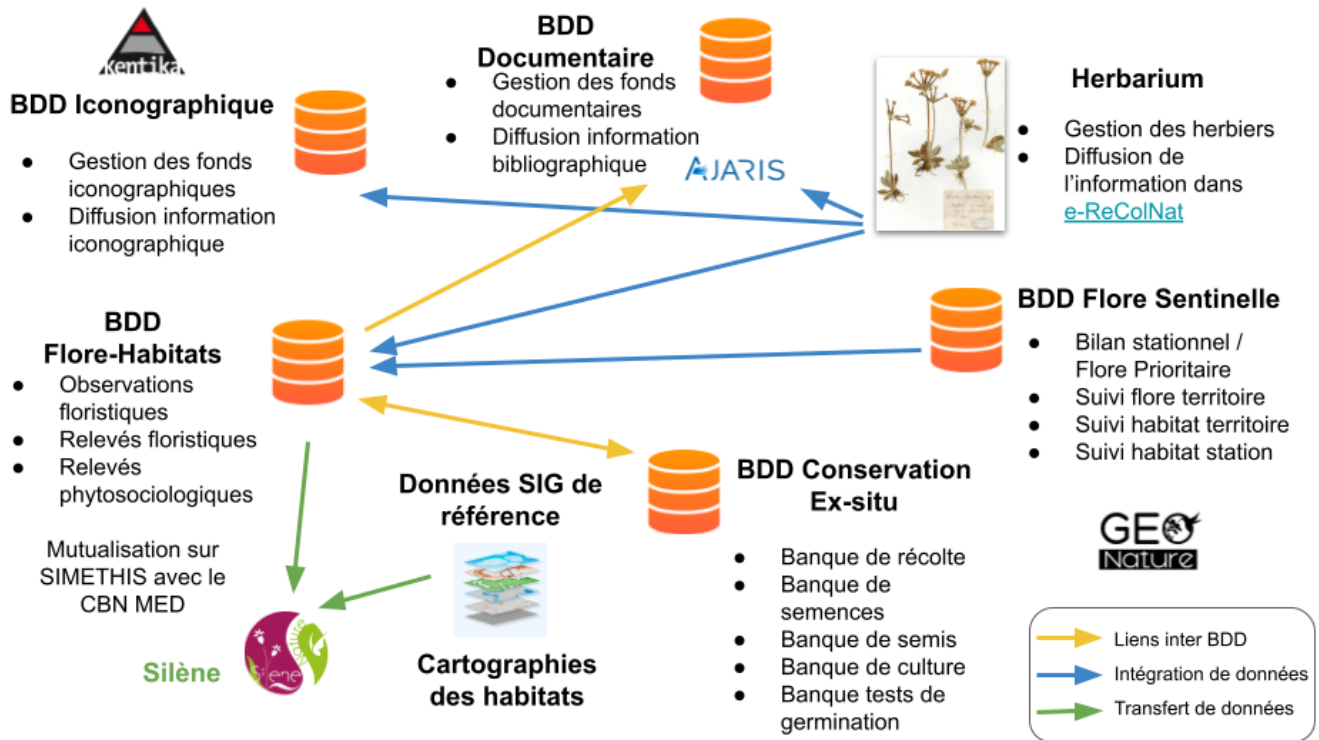


Figure 3 : Les données du CBNA, source : CBNA

Toutes les données du CBNA sont stockées sur 6 serveurs (voir figure 3) :

- 3 hébergés au sein d'un projet nommé "CBNA" de l'offre Public Cloud d'OVH :
 - Un serveur Windows sur lequel sont stockées les bases Kentika (Iconographique) et Ajaris (Documentaire).
 - Un serveur Linux qui héberge le réseau Flore Alpes Ain, ancienne base Flore Sentinelle, où sont stockées les données du module "Flore Prioritaire". Ce serveur a été monté pendant mon stage.
 - Un serveur Linux qui héberge un logiciel de gestion de projets, Gefiproj.
 - Un serveur de sauvegarde de tous les autres serveurs
- 2 hors projet Public Cloud :
 - Un serveur Linux mutualisé et géré par le CBNMED où est stockée la BDD Flore-Habitats dans Simethis, autrement appelé réseau Conservation (voir schéma).
 - Un serveur pris dans le cadre de l'offre VPS (modèle SSD-3) qui héberge actuellement le projet Flore Sentinelle.

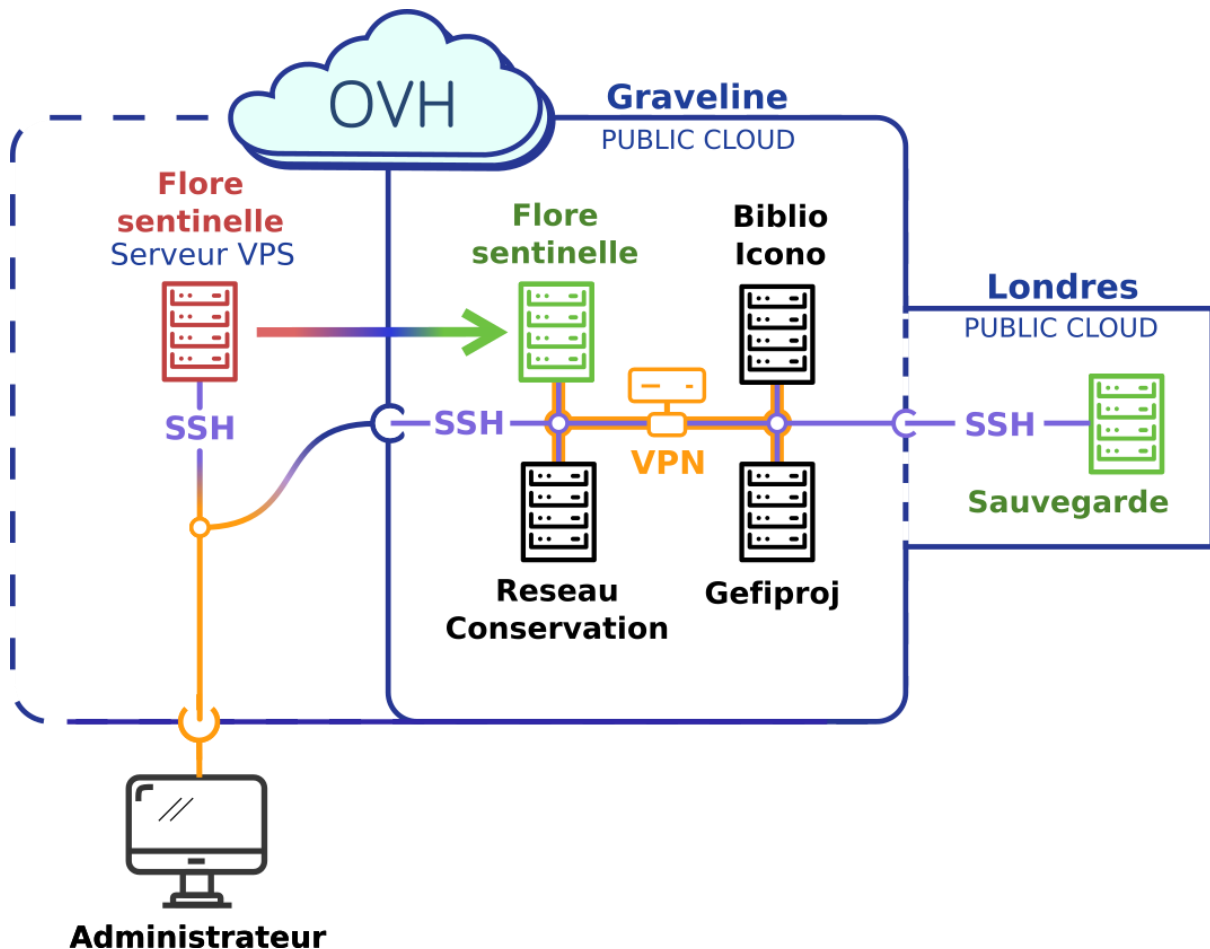


Figure 4 : Les serveurs de données du CBNA, source : Nicolas Rouyer

En plus de ces serveurs, le CBNA dispose à son siège (Gap) de disques durs sur lesquels sont stockés les herbiers numérisés ainsi que deux NAS (serveurs de fichiers) où sont stockées les données SIG de références ainsi que les cartographies des habitats.

Le serveur qui hébergeait la base Flore Sentinelle était jusqu'à présent peu sécurisé et difficilement maintenable dans le temps. De plus, il était arrivé à ses limites, c'est pourquoi le CBNA a souhaité le remplacer par un nouveau serveur qui sera hébergé dans le même projet Public Cloud que les autres serveurs. Cette solution permet d'offrir plus de scalabilité au nouveau serveur, la formule actuellement utilisée (VPS SSD-3) étant à son maximum de potentiel et n'offrant pas de possibilité d'évolution. Ce nouveau serveur sous Linux accueillera la BDD Flore Sentinelle et l'ancien serveur sera résilié.

Afin de renforcer la sécurité des données du CBNA, un serveur de sauvegarde a été créé sur le même Public-Cloud, mais basé à Londres. Il servira à sauvegarder les autres serveurs basés à Gravelines. Il fera office d'espace de stockage temporaire afin de télécharger les images (snapshots) des serveurs, c'est-à-dire des instantanés de l'état des données des serveurs. Il servira également à stocker les sauvegardes des bases de données et les dossiers systèmes des autres serveurs dans des dépôt Borg (outil de sauvegarde incrémentielle et sécurisé), où les données transiteront jusqu'au serveur de sauvegarde via SSH.

Le montage du nouveau serveur Linux accueillant la BDD Flore Sentinelle ainsi que le montage du serveur de sauvegarde sont effectués en parallèle de mon stage par Nicolas ROUYER, un autre stagiaire au service SI du CBNA. Cette opération doit permettre à long terme de supporter les futures montées en charge du SI, faciliter l'administration des serveurs et améliorer la sécurité des applications présentes.

Enfin, depuis 2019, le développeur du CBNA a mis en place un DokuWiki collaboratif, installé sur un hébergement mutualisé et accessible depuis le web. Son accès est limité en écriture et lecture par mot de passe seulement aux agents du CBNA. Il sert d'espace de stockage en ligne pour toute information relative au fonctionnement du CBNA, l'équipe, les projets, les équipements... Chaque service a également son espace réservé et c'est sur cette plateforme que toutes les procédures et documentations sont partagées. Dans le même principe, un dépôt GitHub¹⁴ dédié aux développements du CBNA a été créé.

GEONATURE

L'environnement applicatif

GeoNature est un ensemble d'applications web et mobiles pour saisir, gérer, synthétiser et diffuser des données faune, flore et fonge. La première version de cet environnement applicatif a été développée entre 2010 et 2015 par les services SI du PNE et du PNC (Cévennes). Sa deuxième version est en cours de développement depuis 2017, la version stable actuelle étant la 2.9.2. C'est un outil open source développé et maintenu par un collectif d'une dizaine de développeurs issus du réseau des Parcs Nationaux et de quelques partenaires comme le CBNA.

- **Le fonctionnement**

GeoNature permet de déployer un système d'informations complet pour la gestion des données naturalistes, donnant accès à :

- La gestion des référentiels taxonomiques et utilisateurs (TaxHub¹⁵ et UsersHub¹⁶)
- La gestion des nomenclatures qui répondent aux standards de l'INPN
- La saisie web et mobile de données dans différents protocoles
- La gestion des métadonnées
- L'intégration des données de partenaires
- L'export des données selon les formats attendus par chaque partenaire
- La synthèse des données des différents protocoles (module Synthèse)
- La diffusion des données sur un portail web grand public (GeoNature-Atlas)

¹⁴ <https://github.com/cbn-alpin>

¹⁵ <https://github.com/PnX-SI/TaxHub>

¹⁶ <https://github.com/PnX-SI/UsersHub>

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 des droits d'un utilisateur)
- L'API de GeoNature (get, post, update des données des différents modules, métadonnées, intersections géographiques, exports...)

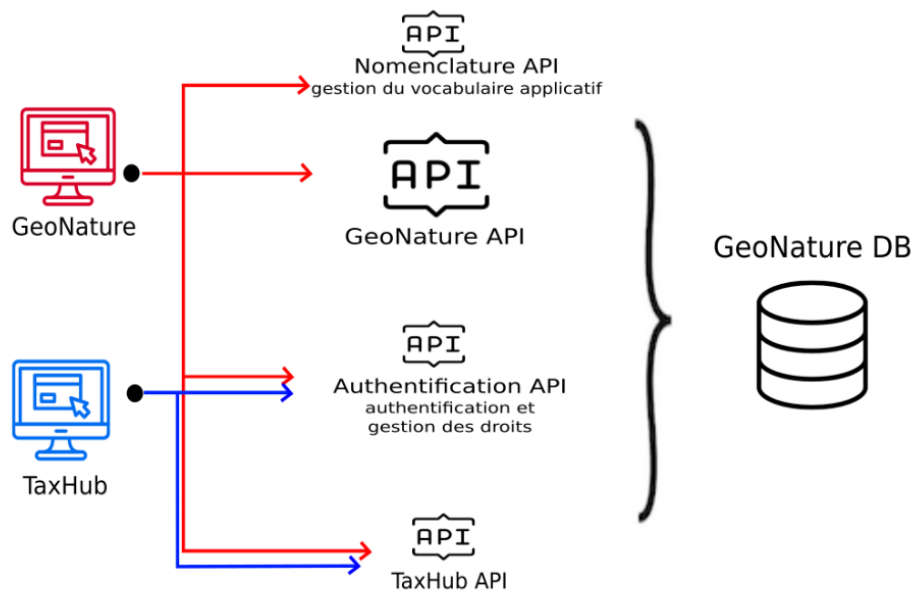


Figure 5 : Fonctionnement de GeoNature, source : documentations officielle GeoNature

• Les technologies

Initialement, GeoNature a été développé avec PHP, Symfony et ExtJS. En 2017, les parcs nationaux français ont décidé de refondre GeoNature complètement avec une nouvelle version (V2) réalisée en Python/Flask/Angular.

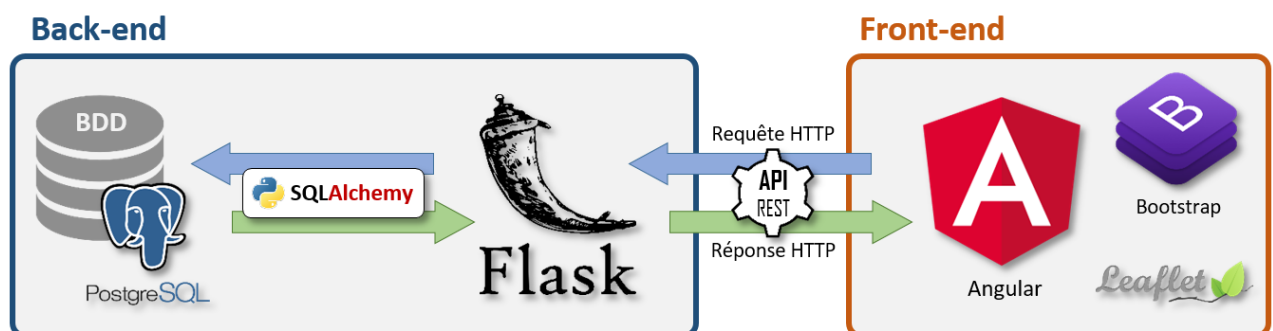


Figure 6 : Les technologies de GeoNature, source : documentation officielle de GeoNature

GeoNature comme outil métier au CBNA

Le CBNA a fait le choix d'utiliser GeoNature¹⁷ comme outil métier du service Conservation, par rapport à sa fiabilité et sa modularité.

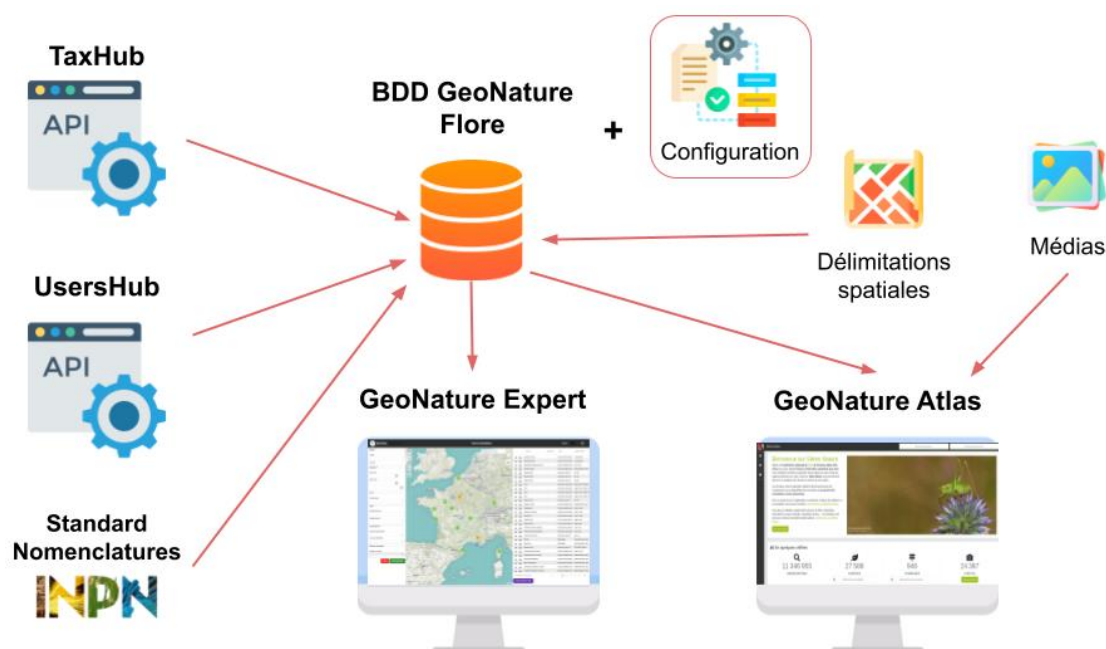


Figure 7 : Fonctionnement de l'environnement applicatif GeoNature au CBNA, source : Cendrine HOARAU

De plus, GeoNature joue aujourd'hui un rôle central dans le déploiement de projets nationaux tels que DEPOBIO (Dépôt Légal de données brutes de Biodiversité). Cette instance créée dans le cadre de la loi pour la reconquête de la biodiversité du 8 août 2016 repose techniquement sur GeoNature pour agréger l'ensemble des données collectées par les bureaux d'études pour les études d'impact d'aménagements. GeoNature est également utilisé comme outil technique dans le cadre du projet ministériel GINCO (Gestion d'Information Naturaliste Collaborative et Ouverte), partie intégrante de la stratégie nationale de mise en œuvre des SINP (Système d'Information de l'iNventaire du Patrimoine naturel). Les enjeux de ces projets sont d'améliorer l'interopérabilité des données et des systèmes d'information répartis.

GeoNature étant conçu de manière générique et ouverte, il permet à une structure comme le CBNA, de récupérer l'essentiel du fonctionnement et de développer des modules pour les besoins métiers spécifiques de ses agents. De plus, de par sa proximité géographique avec le siège du PNE, il est plus facile d'échanger avec l'équipe qui pilote GeoNature sur les développements en cours. La mutualisation des développements est par ailleurs intéressante puisque chaque investissement d'une structure bénéficie au collectif. Enfin, le développeur du

¹⁷ Demo GeoNature : <https://demo.geonature.fr/>

CBNA étant pleinement impliqué dans la gestion des plateformes SINP PACA et AURA qui reposent sur GeoNature, il dispose déjà des compétences techniques requises pour déployer et maintenir cet environnement applicatif au CBNA.

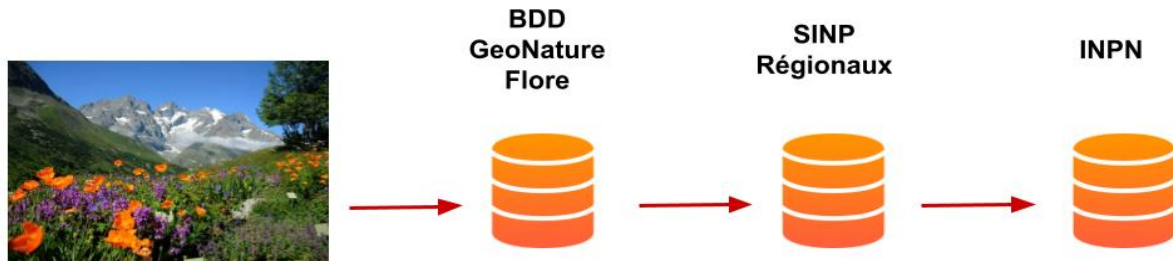


Figure 8 : Cheminement des données du CBNA depuis le terrain jusqu'à la plateforme nationale de l'INPN, source : Cendrine HOARAU

Ce sont tous ces avantages qui pourraient contribuer sûrement à terme à faire de GeoNature, le futur outil métier du CBNA. Cependant, jusqu'à présent, les agents du CBNA utilisent l'application web SImethis, détenue par le CBNMed. Même si elle répond bien aux besoins métiers du CBNA, les technologies utilisées par cet outil, notamment JQuery, peuvent s'avérer difficiles à maintenir avec la multiplication des fonctionnalités du projet. De plus, le code source n'est pas ouvert, ni rendu facilement disponible par l'intermédiaire d'un gestionnaire de versions public (GitHub). De ce fait, l'objectif est qu'à terme, le service SI du CBNA développe des modules GeoNature correspondants à chacun de ses protocoles, pour que toute la chaîne de travail du CBNA soit homogène et que la structure puisse devenir indépendante techniquement.

LES OBJECTIFS DU STAGE

Les objectifs du stage ont évolué à mon arrivée au CBNA. En effet, il était convenu que le stage commence par la migration des 4 modules Conservation vers la nouvelle version de l'environnement applicatif GeoNature. Cependant, une opportunité de travailler sur l'outil cartographique de GeoNature, le module Synthèse, s'est présentée. Le travail devait être réalisé avant mon arrivée en stage et ne pouvait donc pas servir de sujet de stage. Mais finalement, à la suite de retards, ce travail a pu être effectué durant le stage et venir avantageusement remplacer la seconde mission initialement prévue. Ce travail est une demande du SINP PACA auprès du CBNA pour améliorer l'affichage des données sur la Synthèse de Silene Expert, leur outil métier GeoNature.

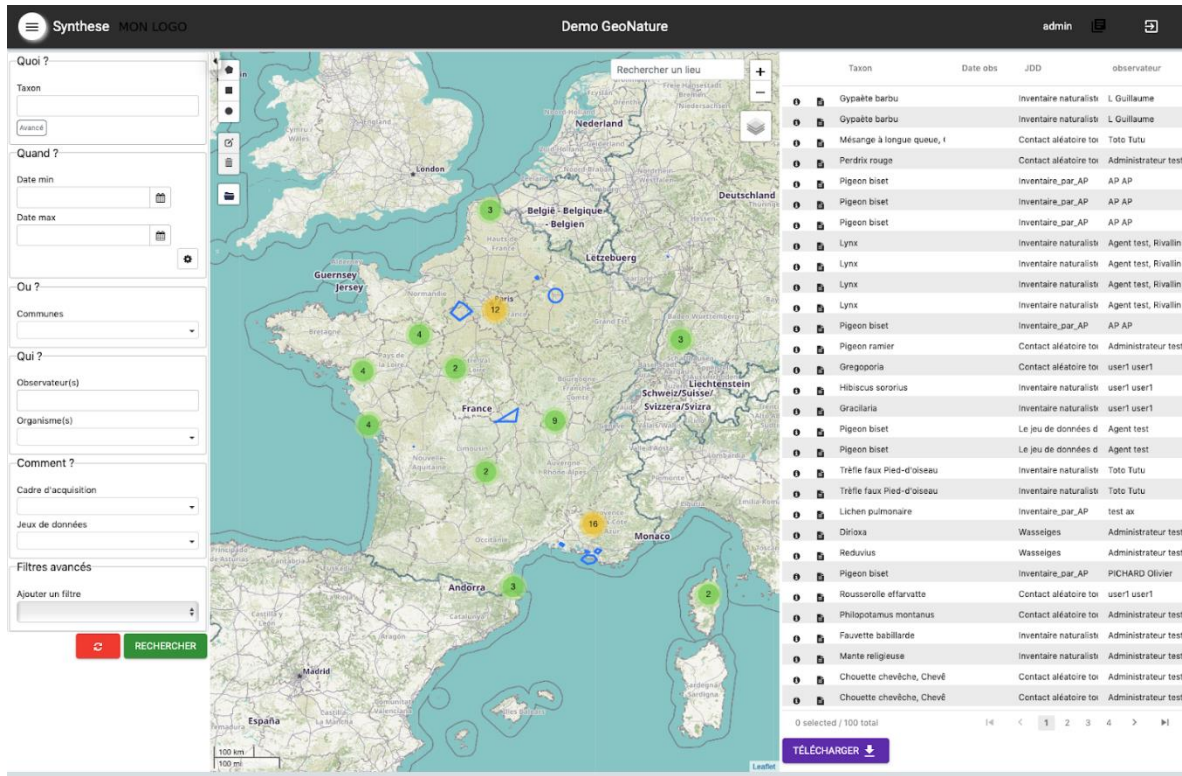


Figure 9 : Le module Synthèse de GeoNature Expert, source : GeoNature

La Synthèse est un outil métier qui permet aux agents de requêter des données naturalistes et de pouvoir les exporter ensuite. Au centre de cet outil, une carte Leaflet, avec à gauche un panneau de filtres et à droite de la carte, une liste déroulante correspondant aux résultats de la recherche (voir figure 8). Jusqu'à présent, l'affichage des données (observations naturalistes) de la Synthèse se présentait sous forme de cluster de points. Plusieurs utilisateurs ont fait remonter le besoin de disposer également d'un affichage des données en mailles colorées en fonction du nombre d'observations qu'elles contiennent, afin d'avoir une meilleure vue d'ensemble de la répartition des espèces sur le territoire. La première mission de stage a donc été de réaliser l'implémentation d'un bouton sur la carte Leaflet de la Synthèse GeoNature, permettant de basculer d'un affichage en mode points (*cluster*) ou en mode maille. Ce développement nécessite une compréhension globale de GeoNature et du fonctionnement de la Synthèse, ainsi qu'une montée en compétence rapide sur Python, SQLAlchemy et Angular (Javascript et Typescript).

La seconde mission du stage correspond à la montée en version des 4 modules GeoNature de Conservation de la flore, et à leur migration sur le nouveau serveur Linux du réseau Flore Sentinelle. Pour ce faire, l'objectif dans un premier temps est de monter en version GeoNature, de la version 2.4.1 vers la dernière version stable disponible (2.9.2). Cela implique de s'assurer que la migration des données dans la base Postgresql s'effectue correctement. Cette mise à jour nécessitera aussi d'adapter les 4 modules de conservation au nouveau format de module disponible depuis la version 2.8.0, c'est-à-dire au format packagé (voir page 37). Lorsque l'ensemble sera fonctionnel, il sera nécessaire de déployer le tout sur le nouveau serveur du réseau Flore Sentinelle.

AMÉLIORATION DE GEONATURE EXPERT

MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL

Afin de pouvoir débiter les missions de stage, la première étape a été d'installer sur mon poste de travail, un PC sous Linux Debian 11, la dernière version de GeoNature. La première mission consistant à améliorer la visualisation des données sur la Synthèse de l'outil métier du SINP PACA, Silene Expert, les développements ont été effectués sur la dernière version de GeoNature en développement, la 2.10.

Pour mener à bien cette première étape, une prise en main de GitHub a été nécessaire. En effet, GeoNature étant un outil open source collaboratif, tous les développements en cours sont en ligne, sur le dépôt GitHub de GeoNature. A chaque nouvelle fonctionnalité ajoutée, une branche de travail Git est créée. Ainsi, la branche "feat/meshes" ("mailles" en anglais) est la branche sur laquelle j'ai fait les développements liés à ma première mission de stage. Une fois les développements terminés et validés par la communauté GeoNature, cette branche a été adaptée pour être incluse dans la branche de travail du SINP PACA ("feat/sinp").

L'objectif a donc été de reproduire une installation de Silene Expert sur mon PC afin de commencer à développer. Pour cela, un compte professionnel Github (GH), membre de l'organisation "cbn-alpin"¹⁸ a été créé. Puis, le dépôt GeoNature a été "forké" sur mon dépôt GH. Un Fork est une copie d'un dépôt, sur un dépôt distant (voir figure 9). Le plus souvent, les forks sont utilisés pour proposer des modifications au projet d'un tiers auquel l'accès est limité à l'écriture. Le travail s'effectue ensuite en local, sur le clone du Fork (dépôt local). Une fois les développements terminés, une "pull request" (demande d'extraction) est envoyée sur le dépôt référent pour acceptation.

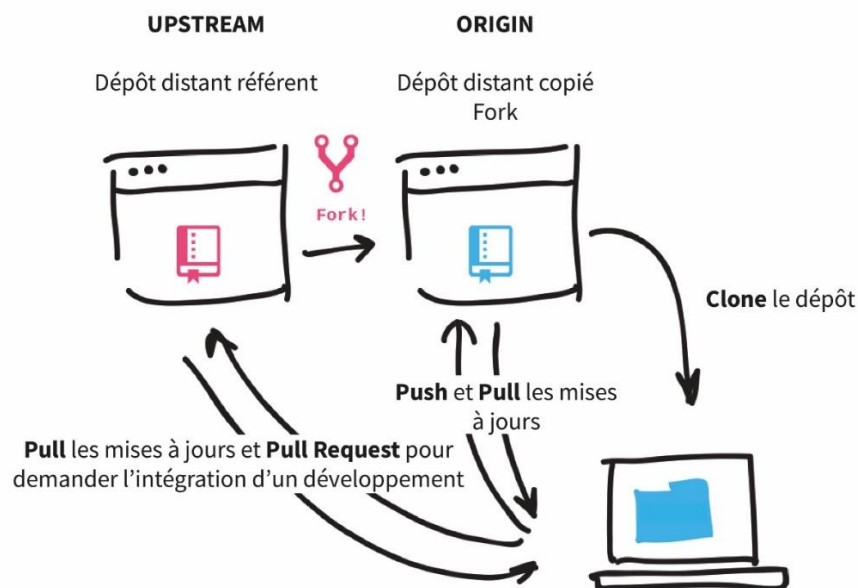


Figure 10 : Fonctionnement des dépôts Github, source : Cendrine HOARAU

¹⁸ <https://github.com/cbn-alpin>

Puis, les dépôts de TaxHub (gestion des taxonomies) et de UsersHub (gestion des utilisateurs) ont été clonés en local sur mon PC. Puisqu'aucune modification n'est envisagée sur ces deux modules, nous n'avons pas besoin de "forker" le dépôt.

Une fois les dépôts GN, TH et UH récupérés en local, il faut installer PostgreSQL avec son extension PostGIS et récupérer la base de données GeoNature du SINP PACA. Pour cela, un dump de la BDD, c'est-à-dire un fichier qui contient la structure et le contenu de la BDD du SINP PACA, a été installé sur ma machine.

Enfin, la documentation officielle¹⁹ de GeoNature a été suivie pour réaliser l'installation complète de l'espace de travail. Cette documentation a été complétée par le développeur du CBNA sur le DokuWiki en interne, ce qui m'a permis d'avoir assez de clés en main pour finaliser l'installation.

Ce premier contact avec GeoNature, GitHub et le fonctionnement du terminal Linux, a été complété par des lectures recommandées par mon maître de stage, afin de me familiariser avec les technologies utilisées par GeoNature ainsi que les bonnes pratiques de développement :

- R. C. Martin et H. Soulard, *Coder proprement*. Paris: Pearson, 2009.
- *Deviens un ninja avec Angular2*, Ninja Squad.
- T. Fredrich, *RESTful Service Best Practices*.

Une fois l'installation de l'environnement de travail finalisé, j'ai pu commencer à travailler sur l'amélioration de la visualisation des données géographiques sur la Synthèse de GeoNature.

AMÉLIORER LA VISUALISATION DES DONNÉES GÉOGRAPHIQUES

L'amélioration de la visualisation des données géographiques sur l'outil central de GeoNature Expert, la Synthèse, consiste à optimiser le fonctionnement de la carte Leaflet selon deux axes. Le premier axe de travail porte sur l'agrégation des données géographiques par géométrie. Le second consiste à ajouter un nouveau mode d'affichage des données, l'affichage par mailles, colorées en fonction du nombre d'observations naturalistes. Ces développements ont été sollicités en début d'année 2022 par les membres du SINP PACA, dans le but à la fois d'optimiser le chargement d'une grande quantité de données sur la carte, et aussi d'avoir une meilleure vue d'ensemble de la répartition des espèces sur le territoire.

¹⁹ <https://docs.geonature.fr/installation.html>

Agrégation des données par géométrie

Jusqu'à présent, la version stable (branche "master") de GeoNature dispose de l'outil Synthèse avec une carte Leaflet où sont renvoyées les observations naturalistes sous forme d'objets géographiques (points, lignes, polygones) clusterisés. Cette fonctionnalité est rendue possible grâce à la librairie JS Leaflet.markercluster²⁰, qui agrège les données proches géographiquement par clusters (groupes). La mise en place de ce plugin début 2019, a déjà permis d'améliorer les performances de la Synthèse. En effet, le Synthèse était jusqu'alors limitée à 10 000 observations renvoyées. Grâce aux clusters, la limite d'affichage des données passe à 50 000.

- **Fonctionnement actuel et problèmes liés**

Cependant, cette clusterisation a engendré des difficultés à l'utilisation de la carte. En effet, deux problèmes majeurs ont souvent été reportés, celui de la perte d'information visuelle quant à la répartition des données sur le territoire, et le manque de praticité pour accéder aux informations lorsque les observations ont une géométrie commune ou très proche. De plus, plusieurs utilisateurs ont depuis fait remonter le besoin de relever la limite des 50 000 observations.

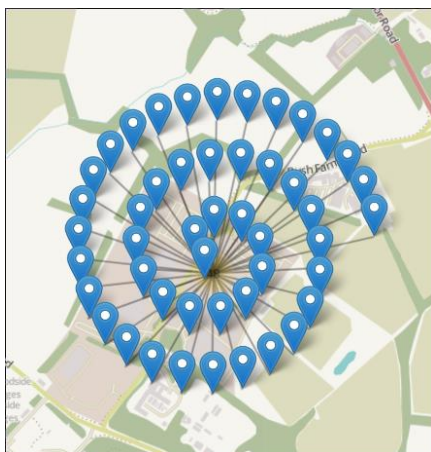


Figure 11 : Spirale de markers
Leaflet, source : Leaflet



Figure 12 : MarkerCluster Leaflet, source : Leaflet

Les observations ayant une géométrie commune ou très proche sont affichées sous forme de spirale à très grande échelle. Ainsi, les observations sont difficilement identifiables par l'utilisateur et celui-ci est obligé de cliquer sur chaque marker pour obtenir des informations relatives.

A petite échelle, les clusters ne permettent pas facilement à l'utilisateur d'avoir une idée de la répartition des observations sur un territoire. Ceci est engendré par une erreur de sémiologie graphique. Le nombre d'observations (variable quantitative discrète) devrait être représenté par un symbole qui varie par sa taille, et non par sa couleur. Les MarkerCluster ont tous la même taille ce qui rend confus la lecture de la carte.

²⁰ <https://github.com/Leaflet/Leaflet.markercluster>

Afin de pallier ces problèmes, un développeur de la communauté GeoNature a réalisé en début d'année un développement qui permet d'agréger les données par géométrie, travail préparatoire à l'agrégation des données par mailles. Cependant, la manière dont les données ont été agrégées n'était pas optimale. En effet, les valeurs relatives à chaque observation étaient réparties dans des tableaux affectés à chaque propriété. Pour reconstruire l'objet représentant une observation, il fallait par exemple récupérer pour chaque propriété les valeurs de l'index 1 des tableaux correspondants (voir figure 12). De plus, ce format de données ne respectait pas les bonnes pratiques du format JSON, pour lequel il faudrait utiliser des fonctions d'agrégation telles que "json_agg" et "json_build_object" pour contenir les propriétés de l'objet.

Par ailleurs, l'agrégation des données était réalisée dans un second temps en python, après avoir récupéré toutes les données dans la requête SQL initiale. Afin de fluidifier la récupération des données, l'agrégation des données par géométrie se fera dans la requête SQL initiale, puis la route Flask renverra une collection de features GeoJSON.

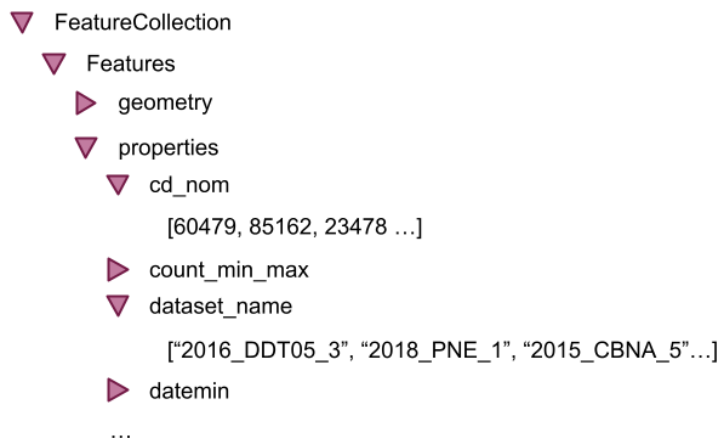


Figure 13 : Ancien format de récupération des données de la Synthèse, source : Cendrine Hoarau

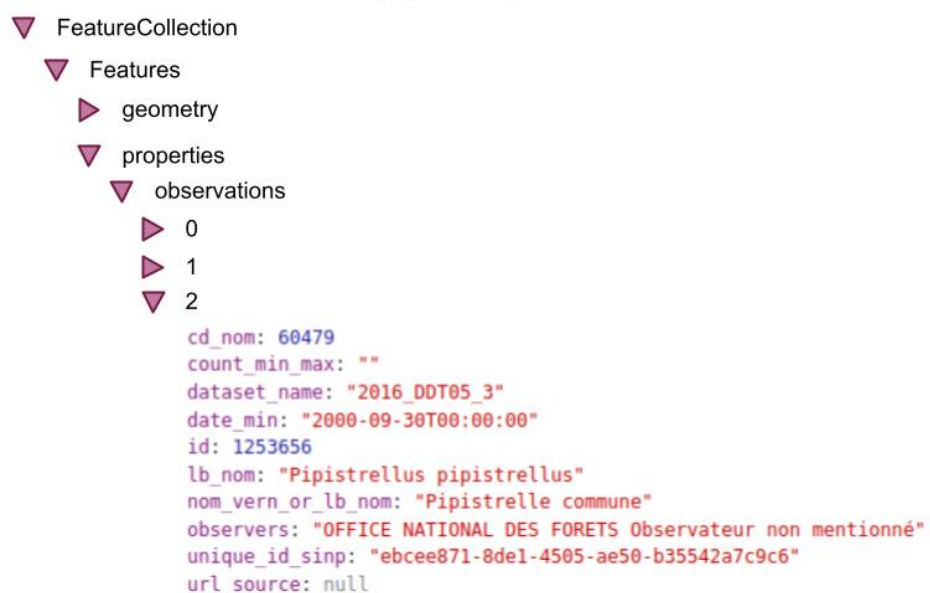


Figure 14 : Nouveau format de récupération des données dans la Synthèse, source : Cendrine HOARAU

Amélioration du backend

- **Modification de la récupération et de l'agrégation des données**

Ainsi, la première étape a été de reprendre ce développement. Pour commencer, la requête SQL initiale a été extraite du code Python et récupérée dans le terminal lors de son exécution. Le travail de modification a ensuite été réalisé dans DBeaver, le logiciel libre SQL utilisé au CBNA (voir annexe 1). Les géométries ainsi que toutes les informations utiles liées aux observations sont récupérées dans une requête CTE²¹ (Common Table Expressions), ce qui permet de fixer une limite d'observations renvoyées, paramètre modifiable dans les fichiers de configuration. Puis, ces informations sont récupérées dans la requête suivante qui construit l'objet GeoJSON (geometry, properties). Ce GeoJSON est ensuite récupéré dans la route Flask, sous forme de collection d'objets GeoJSON (FeatureCollection).

Chaque entrée du résultat de cette requête a pour géométrie les coordonnées GeoJSON de la requête CTE initiale (alias "observations"), et pour propriété une agrégation d'objets JSON (*json_agg* et *json_build_object*) avec les informations spécifiques à chaque observation. Cette agrégation prend effet suite au regroupement final (*group by*) des données par géométrie.

Cette modification (voir figure 13) a permis d'optimiser le temps de chargement des données sur l'application. Celui-ci est d'autant plus performant que le nombre de données requêtées est important, mais la différence de performance n'excède pas plus de 1 seconde l'ancien fonctionnement. Par ailleurs, les objets GeoJSON peuvent à présent être typés avec TypeScript dans le frontend (Angular) ce qui n'était pas le cas avant. Les types sont un moyen de distinguer les programmes corrects des programmes incorrects avant de les exécuter, en décrivant dans le code comment est prévue l'utilisation des données.

- **Modification de la route Flask**

Une fois la requête testée et approuvée, la seconde étape était de traduire cette requête en langage Python à l'aide de SQLAlchemy. SQLAlchemy est une boîte à outils open source SQL et un ORM²² (Object-Relational Mapping). Cette sur-couche au SQL permet de synchroniser les classes Python du code avec les tables correspondantes en base de données. Cette étape a été plus chronophage étant donné la difficulté du fonctionnement de SQLAlchemy. En effet, certaines erreurs renvoyées et la documentation officielle²³ sont difficilement appréhendables.

Après avoir optimisé la requête SQL de récupération des données pour la Synthèse et modifié la route Flask en conséquence, j'ai proposé un nouvel affichage des observations sur la carte.

²¹ <https://docs.postgresql.fr/11/queries-with.html>

²² <https://definitions-digital.com/developpeur/orm>

²³ <https://www.sqlalchemy.org/>

Amélioration du frontend



Les observations étant à présent regroupées par géométrie, il n'est plus correct de n'afficher qu'un marqueur (*circleMarker*) s'il y a plusieurs observations (voir ci-contre). J'ai donc ouvert un ticket²⁴ sur GitHub pour suggérer à la communauté GeoNature une nouvelle visualisation.

Le style actuel du *circleMarker* ne permettant pas à l'utilisateur de se rendre compte du nombre d'observations sur une même géométrie, j'ai proposé de faire varier le rayon de ce *circleMarker* de manière proportionnelle au nombre d'observations et d'afficher en son centre le nombre d'observations (comme pour l'affichage en cluster).



Figure 15 : Proposition de style pour les *CircleMarker*, source : Leaflet



Figure 16 : Style des *CircleMarker* utilisé dans la Synthèse, source : Leaflet

Après discussion, la variation du rayon du *circleMarker* en fonction du nombre d'observations contenues ne paraît pas pertinent puisqu'une géométrie peut contenir des centaines d'observations ce qui entrerait en conflit visuel avec l'affichage des *markerCluster*. De ce fait, nous n'avons pas retenu cette proposition. Les autres développeurs ont cependant accepté la proposition d'afficher au centre du *circleMarker* le nombre d'observations.

Ce premier développement m'a permis de voir et comprendre le fonctionnement de GeoNature, ainsi que de m'initier aux nouvelles technologies telles que SQLALchemy côté backend et Angular côté frontend.

Angular est un framework open source, basé sur du TypeScript, un langage qui sur-couche le JavaScript dans le but de sécuriser le code et permettre notamment le typage²⁵. Ce framework est basé sur une architecture du type MVC (Modèle-vue-contrôleur) et permet donc de séparer les données (modèle), le visuel (interface graphique) et les actions de l'utilisateur.

²⁴ <https://github.com/PnX-SI/GeoNature/issues/1847>

²⁵ typage : définir la nature des valeurs que peut prendre une donnée.

Ces premiers pas avec les technologies de GeoNature m'ont permis de réaliser par la suite l'ajout d'une nouvelle fonctionnalité sur le module Synthèse : l'affichage par mailles, colorées en fonction du nombre d'observations.

Affichage des données par mailles

- **Besoins**

Suite à plusieurs retours utilisateurs, les membres du SINP PACA ont fait remonter le besoin de disposer d'un nouveau mode de visualisation des observations dans la Synthèse : le mode mailles, tel qu'il existe sur le portail grand public, Silene Nature²⁶ (GeoNature Atlas).

Cependant, contrairement à ce qui a été fait sur l'Atlas, où le mode mailles bascule en mode points lorsque l'utilisateur zoome sur la carte, un bouton (*slide toggle*) a été ajouté dans la carte de la Synthèse afin que l'utilisateur choisisse le moment où il bascule du mode mailles au mode points. Cette décision a été prise en premier lieu parce que le passage du mode mailles au mode points lors d'un zoom (et d'un dézoom) nécessite un certain temps de chargement côté client (plusieurs secondes). Puis, pour garder le fonctionnement de la Synthèse actuelle avec les MarkerCluster, il paraît pertinent de séparer les deux modes d'agrégation des observations : les mailles / les MarkerCluster. Enfin, cette fonctionnalité sera intégrable au module Synthèse de Silene Expert qui fonctionne avec le floutage des données sensibles et privées (voir précisions p. 25).

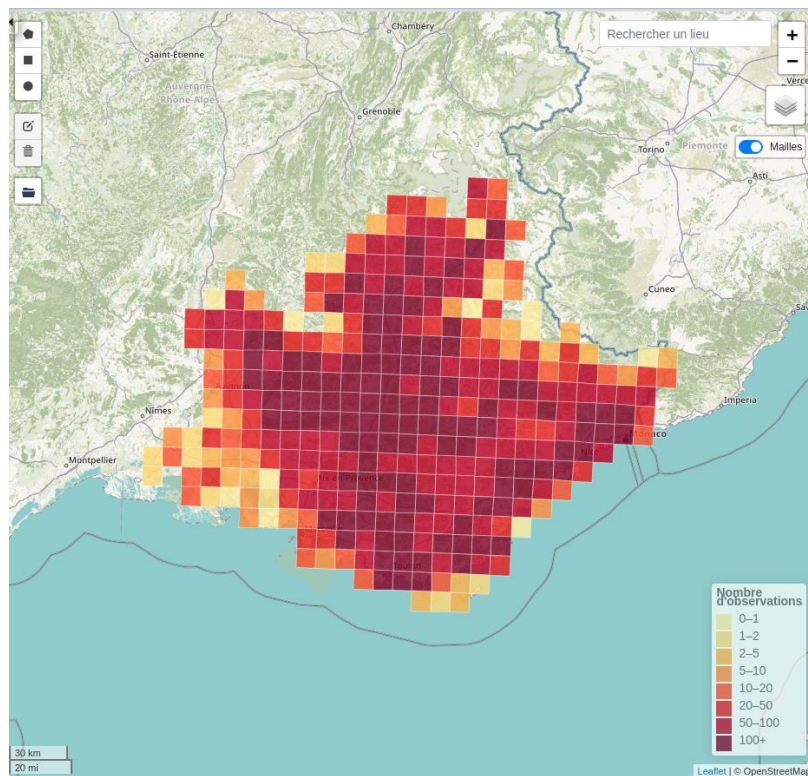


Figure 17 : Affichage par maille dans la Synthèse GeoNature, source : GeoNature

²⁶ <https://nature.Silene.eu/espece/190352>

- **Fonctionnement**

Lorsque l'on clique sur le bouton, l'API *for_web* est appelée en utilisant les filtres actuellement sélectionnés dans le formulaire de Recherche, à gauche de la carte dans la Synthèse. Si on passe au mode mailles, la géométrie des mailles est renvoyée, si on passe au mode Points, c'est la géométrie des points qui est renvoyée, tout en gardant le fonctionnement des MarkerCluster. Lorsque le mode mailles est activé, la légende apparaît, puis disparaît lorsqu'on passe en mode Points. Lorsque le mode mailles est activé, que l'on zoome sur une maille et que l'on passe en mode Points, le niveau de zoom reste le même. Au contraire, lorsque l'on passe du mode Points au mode mailles, la carte dézoome jusqu'à la boundingbox des résultats de la Recherche.

- **Configuration**

Des paramètres de configuration ont été créés afin d'activer ou non ce mode mailles dans la Synthèse :

- *ENABLE_AREA_AGREGATION* : ajout du bouton et donc activation de la fonctionnalité.
- *AGREGATION_AREA_TYPE* : choix du zonage pour l'agrégation des observations (Maille 5x5km / Maille 10x10km / Commune...).
- *AREA_AGREGATION_BY_DEFAULT* : choix de la position du toggle activée par défaut (mode mailles ou Points).
- *AREA_AGREGATION_LEGEND_CLASSES* : ajustement des classes pour garder une sémiologie graphique cohérente en fonction de *AGREGATION_AREA_TYPE*.

- **Backend**

Dans la route *for_web* de l'API, le paramètre *with_areas* a été ajouté aux filtres de la requête de récupération des données. Ainsi, lorsque ce paramètre est à *True*, la requête récupère les géométries des mailles correspondants aux observations. Cela est rendu possible grâce à une table de liaison entre la table contenant les mailles (*l_areas*) et la vue contenant les identifiants des observations (*v_synthese_for_web_app*).

Étant donné que la requête de récupération des données regroupe désormais les résultats par géométrie, chaque maille ne sera renvoyée qu'une fois. Ainsi, l'intégration de cette nouvelle fonctionnalité côté backend a nécessité peu de changement grâce au travail précédent d'agrégation des données par géométrie.

Cependant, cette fonctionnalité étant destinée à être intégrée dans le GeoNature du SINP PACA, Silene Expert, il était nécessaire de faire évoluer davantage son fonctionnement pour qu'il prenne en compte le floutage des données privées et sensibles, déjà en place. En effet, certains partenaires ne souhaitent pas diffuser au grand public la localisation exacte de leurs

données, comme c'est le cas par exemple pour des espèces protégées. Dans ce cas, si les utilisateurs veulent accéder à des observations géo-confidentielles, ils doivent faire une demande de permission via un formulaire directement accessible sur l'interface GeoNature via leur profil utilisateur.

- **Mailles et floutage des données**

Le floutage des données consiste à afficher une maille sur la carte, à la place d'un point (circleMarker). Ainsi, il n'est pas possible d'obtenir la localisation exacte de l'observation, mais celle-ci est quand même présente dans les résultats de la requête. Dans Silene Expert, les données privées sont floutées à la maille 5x5km (M5) et les observations sensibles sont floutées à la maille 10x10km (M10). Ainsi, pour un utilisateur ne disposant pas de droits d'accès aux données privées et sensibles et lorsque l'affichage est en mode Points, les observations sur la carte de la Synthèse se présentent sous forme de points clusterisés pour les observations accessibles et sous forme de mailles pour les observations non accessibles (voir figure 17).

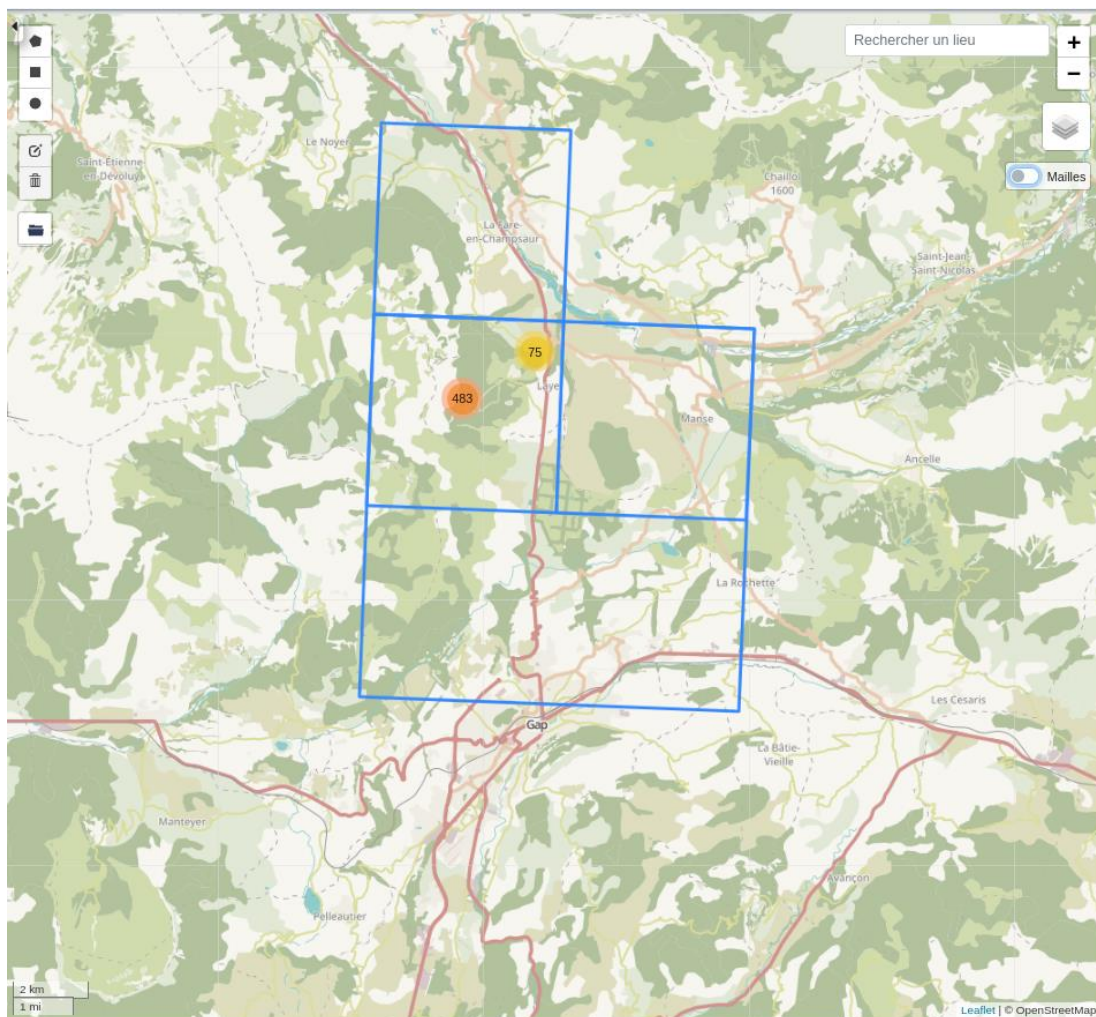


Figure 18 : Affichage par points dans la Synthèse GeoNature avec le floutage des données, source : GeoNature

La configuration de Silene Expert exige un affichage en mode mailles avec des mailles 5x5km. Ainsi, si l'utilisateur requête des données sensibles, floutées à la maille 10x10km, et que le mode mailles est activé, il ne pourra ni distinguer visuellement cette information ni accéder à celle-ci en cliquant dessus, puisque les M5 et les M10 sont imbriquées.

Pour pallier ce problème, lorsque l'affichage Maille est activé, la taille des mailles récupérées ne pourra pas excéder la taille définie dans les paramètres de configuration (*AGREGATION_AREA_TYPE*). Cependant, il n'est pas non plus envisageable de dégrader le niveau de floutage exigé. Ainsi, la solution retenue actuellement dans le cas où la taille de la maille floutée est supérieure à la taille de la maille configurée avec *AGREGATION_AREA_TYPE*, est de récupérer une des mailles qui intersecte la maille supérieure. Par exemple, si l'affichage en mode mailles se fait avec des mailles M5 et que l'utilisateur requête une donnée sensible, et qu'il n'a pas de droit d'accès à ces données, la donnée sensible sera renvoyée dans une des mailles M5 qui intersecte la maille M10 correspondante.

Ainsi, si le paramètre de configuration autorisant le floutage des données (*ENABLE_DATA_BLURRING*) est activé et que l'affichage par maille est également activé alors les opérations suivantes se produisent lors d'une requête :

1. Récupération des mailles qui seront colorées pour les observations requêtées. La taille des mailles est définie dans le paramètre de configuration *AGREGATION_AREA_TYPE*.
2. Récupération des mailles de floutage 5x5km pour les données privées
3. Récupération des mailles de floutage 10x10km pour les données sensibles
4. Vérification des droits d'accès de l'utilisateur, si besoin annule la récupération des mailles de floutage concernées
5. Si des observations doivent être floutées à un niveau égal ou supérieur au paramètre de configuration *AGREGATION_AREA_TYPE*, alors récupération d'une des mailles) à colorer de taille inférieure qui intersecte la maille de floutage supérieure
6. Agrégation des observations par géométrie

En sortie, le résultat de cette requête peut renvoyer des mailles se trouvant à l'extérieur du périmètre de recherche (voir ci-dessous). Ceci est lié à la manière dont les mailles sont récupérées à l'étape 5. Les mailles récupérées qui intersectent la maille de taille supérieure peuvent se trouver à l'extérieur de celle-ci, être adjacentes et non contenues dedans. Pour pallier ce problème il faudrait repartir sur une base de données avec un référentiel de mailles qui s'imbriquent parfaitement. Ce n'est pas le cas avec les mailles fournies dans le cadre de l'INPN (Inventaire national du patrimoine naturel), de ce fait on ne peut pas utiliser de fonction spatiale (*st_Within* ou *st_Contains*) qui récupérerait une des mailles contenues dans la maille de taille supérieure.

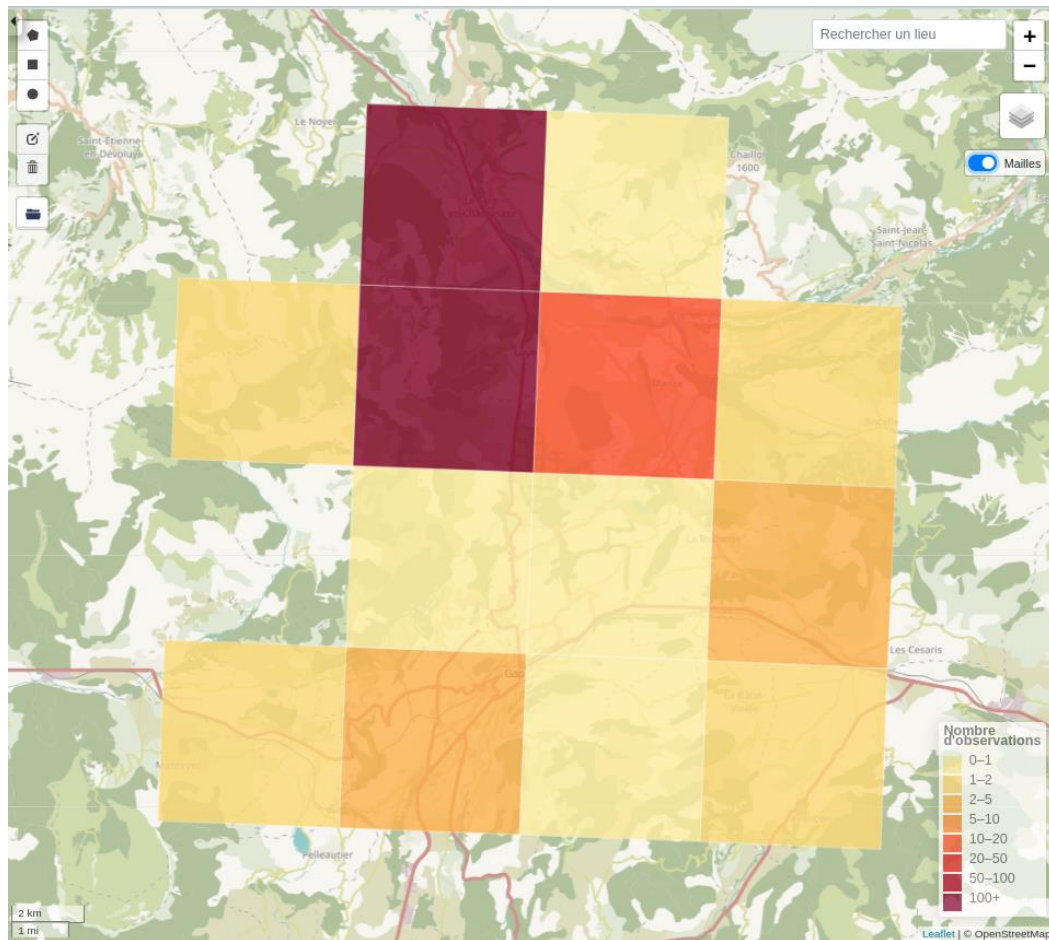


Figure 19 : Affichage par mailles dans la Synthèse GeoNature avec le floutage des données, source : GeoNature

- Frontend

Côté frontend, les fichiers relatifs à la Synthèse se trouvent dans : *geonature/frontend/src/app/syntheseModule*.

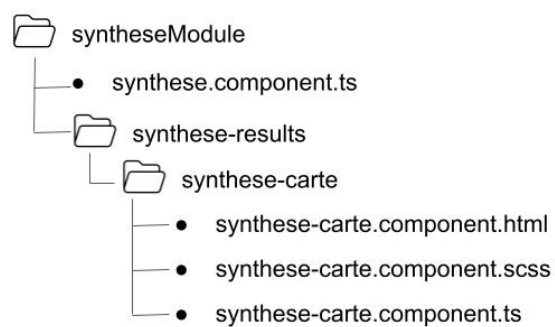


Figure 20 : Répertoires de travail côté frontend pour la Synthèse GeoNature, source : Cendrine Hoarau

Mise en place du bouton

Afin de garder une cohérence visuelle avec les éléments de la carte Leaflet, nous avons utilisé un bouton Leaflet de type switch qui fonctionne comme une checkbox (boîte à cocher).

Si le paramètre de configuration `ENABLE_AREA_AGGREGATION` est activé alors la méthode `addAreasButton` qui permet d'ajouter le bouton à la carte, est appelée. Cette méthode prend effet à l'initialisation de la vue du composant `SyntheseCarteComponent`, dans le hook `@ngAfterViewInit()` :

```
@Component ({
  selector: 'pnx-synthese-carte',
  templateUrl: 'synthese-carte.component.html',
  styleUrls: ['synthese-carte.component.scss'],
  providers: [],
})

export class SyntheseCarteComponent implements OnInit,
AfterViewInit, OnChanges {

  @Output() onAreasToggle = new EventEmitter<any>();

  ngAfterViewInit() {
    // Handle areas button and legend
    if (this.config.SYNTHESE.ENABLE_AREA_AGGREGATION) {
      this.addAreasButton();
      this.onLanguageChange();
      if (this.areasEnable) {
        this.addAreasLegend();
      }
    }
  }
}
```

Extraits de code du fichier `synthese-carte.component.ts`

Un composant (component) est une unité de découpage visuel. Les pages web sont des composants, elles-mêmes découpées en plusieurs composants. L'idée est de séparer les pages web en composants homogènes pour ne pas se contenter de découper par langage (dossiers js/html/css). On donne ainsi une structure logique au projet et à nos fichiers. En termes de code, un composant est une classe Typescript avec des attributs, des méthodes et des constructeurs. Les composants sont déclarés à l'aide de `@Component`.

Un "component" Angular contient :

- Un template contenant l'interface utilisateur en HTML
- Une ou des feuilles de styles associée(s) à ce HTML (SCSS)
- Une classe TypeScript contenant le code associé à la vue, des propriétés et méthodes logiques qui seront utilisées dans la vue

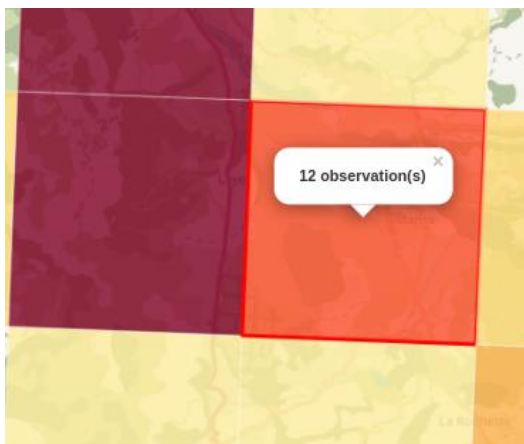
Angular permet d'exécuter du Javascript lors de certains évènements liés au cycle de vie du component. Dans une application Angular, il est utile de pouvoir organiser le code selon ces différentes étapes du cycle de vie (initialisation, évènement, suppression...). Cela est réalisable grâce à ce que l'on appelle des hooks²⁷. La méthode `addAreasButton` est appelée dans le hook `@ngAfterViewInit()`, qui correspond à l'étape suivant l'initialisation de la vue du composant `SyntheseCarteComponent`.

Un "child component" (composant enfant) peut transmettre des données au "parent component" via un mécanisme d' `@Output`. Ces données sont récupérées dans le code HTML via un "listener" qui déclenche une action (event). Ici, le composant `SyntheseCarteComponent` envoie des données à son composant parent `SyntheseComponent` via l' `@Output onAreasToggle`. Cet `@Output` déclenche l'évènement `loadAndStoreData` qui fait appel à l'API de récupération des données `for_web`. Si le bouton des mailles est activé (checked), alors l'API `for_web` renvoie la géométrie des mailles.

```
<pnx-synthese-search
(searchClicked)="loadAndStoreData($event)">
</pnx-synthese-search>
```

Extrait de code du fichier `synthese.component.html`

Coloration des mailles



Afin de rendre compte de la répartition des espèces sur la carte de la Synthèse, les mailles sont colorées en fonction du nombre d'observations qu'elles contiennent. Au clic sur une maille avec la souris, le nombre d'observations contenues s'affiche.

Figure 210 : Zoom sur une pop-up au clic sur une maille dans la Synthèse GeoNature, source : GeoNature

Le nombre d'observations est récupéré en calculant le nombre d'identifiants (id) contenus dans les propriétés du GeoJSON de la maille :

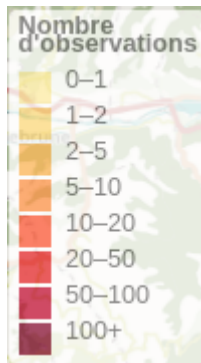
```
private bindAreasPopup(layer, ids) {
  let popupContent = `${ids.length} observation(s)`;
  layer.bindPopup(popupContent).openPopup();
}
```

Extraits de code du fichier `synthese-carte.component.ts`

²⁷ <https://book.angular.fr/composants/untitled>

Si la géométrie du GeoJSON est un polygone et que le bouton des mailles est activé alors le style `setAreasStyle` est appliqué. Ce style fait appel à la fonction `getColor` qui récupère les couleurs des mailles via le paramètre de configuration `AREA_AGGREGATION_LEGEND_CLASSES` (voir Annexe 2).

Légende associée



La discrétisation utilisée pour les mailles de Silene Expert est la même que celle déjà utilisée pour Silene Nature, le portail grand public. Ainsi, les utilisateurs auront le même référentiel quant à la répartition des observations. De plus, malgré le grand différentiel de données qui peuvent être affichées sur la Synthèse, entre 1 et 50 000, cette discrétisation permet de retourner une information de répartition des observations relativement claire.

La légende apparaît sur la carte lorsque le mode mailles est activé, et disparaît lorsqu'on passe en mode Points.

Figure 22 : Légende associée aux mailles sur la Synthèse GeoNature, source : GeoNature

Interaction avec la liste des observations

À chaque clic sur une maille, les observations contenues sont surlignées en vert dans la liste des observations à droite de la carte.

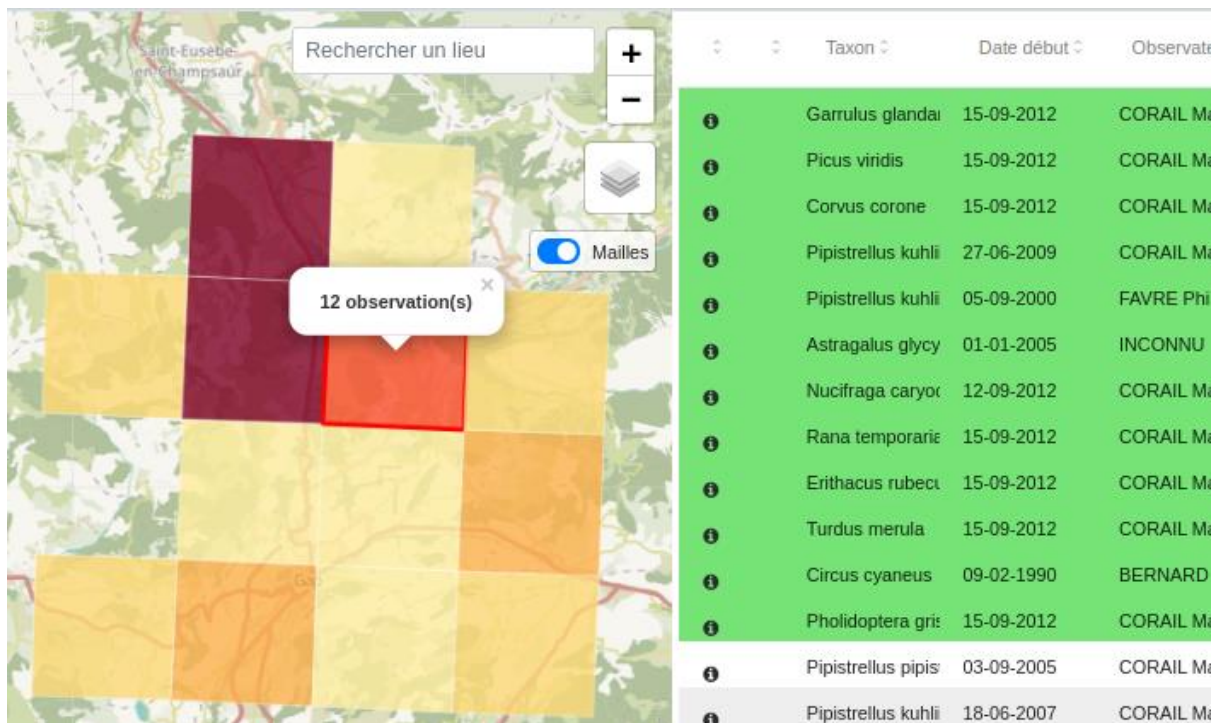


Figure 23 : Observations sélectionnées dans la liste des observations de la Synthèse, source : GeoNature

INTÉGRER LES DÉVELOPPEMENTS

La réalisation de tous ces développements en local a été ponctuée de "commits". Un commit, ou "révision", est une modification individuelle d'un fichier (ou d'un ensemble de fichiers). Lorsqu'un commit est effectué pour sauvegarder le travail réalisé, Git crée un identifiant unique (un "SHA1") qui permet de garder une trace des changements spécifiques effectués ainsi que de l'auteur et de la date de ces changements.

Tous les commits locaux ont été réalisés sur ma branche de travail locale : "feat/meshes". Elle a été créée à partir de la branche distante "develop". Cette branche correspond à la branche de travail principale, c'est elle qui contient les dernières révisions et qui sera à terme fusionnée avec la branche principale "master" afin de monter l'application en version.

Ma branche locale "feat/meshes" a été poussée sur mon Fork, ma copie distante du dépôt Git de GeoNature. Afin d'envoyer mes développements sur le dépôt GeoNature (upstream), j'ai fait une Pull Request²⁸ (PR) (voir figure 9). Les Pull Request, ou demandes d'extraction, permettent d'informer les autres collaborateurs des modifications apportées à une branche d'un dépôt sur GitHub. Une fois qu'une PR est ouverte, il est possible de discuter et revoir les changements potentiels avec les autres développeurs. L'objectif est que les développements soient relus et acceptés afin d'être fusionnés dans la branche "develop" et ainsi être pris en compte dans la future version de GeoNature, la 2.10. Pour l'instant, la PR est encore en attente de relecture.

Par ailleurs, les développements ayant été commandités par le SINP PACA pour leur outil Silene Expert, les développements ont depuis été pris en compte sur leur GeoNature. Cela est possible grâce au fait que les développements leur GeoNature se trouvent sur une autre branche, la branche "feat/sinp", qui évolue pour l'instant en parallèle de la branche "develop".

Fusionner les branches de travail

Afin d'intégrer les développements de la branche de travail "feat/meshes" dans la branche fonctionnelle du GeoNature du SINP PACA, les étapes suivantes ont été réalisées :

1. Squash (agrégation) de tous les commits sur la branche feat/meshes en 1 commit : "Feat synthese: add area aggregation"
2. Création d'une nouvelle branche "feat/meshes-squash" à partir de "feat/meshes"
3. Cherry-pick (récupération) du commit squashé dans une nouvelle branche "feat/sinp-areas" créée à partir de "feat/sinp"
4. Merge (fusion) des développements de "feat/sinp-areas" et résolution des conflits

²⁸ <https://github.com/PnX-SI/GeoNature/pull/1881>

Les difficultés rencontrées avec une application collaborative

L'évolution analogue de la branche de travail du SINP PACA (feat/sinp) et de la branche "develop" de GeoNature est source de conflits pour le CBNA et pour le SINP PACA. En effet, cela fait 2 ans que le développeur du CBNA continue de faire évoluer l'outil métier Silene Expert pour répondre aux besoins des utilisateurs, besoins qui ne sont pas forcément les mêmes pour le reste des utilisateurs de GeoNature.

Cette situation est problématique pour le développeur du CBNA qui est obligé de maintenir la branche "feat/sinp" pour qu'elle soit à jour par rapport à la branche "develop", c'est-à-dire, par rapport aux derniers avancements de l'application. Maintenir une branche de travail sur une application où collaborent une dizaine de développeurs est très complexe et chronophage puisqu'il faut à la fois que les développements faits sur chaque branche puissent fonctionner ensemble et suivre de manière régulière tous les développements en cours. Dans notre cas, la branche "feat/sinp" a 429 commits de retard sur la branche "develop". Le travail de récupération de ces commits peut prendre plusieurs semaines. Hors, le développeur du CBNA est le seul à avoir les capacités techniques pour faire ce travail au sein de la structure.

Cette problématique est sous-jacente à une autre problématique, celle de la gouvernance de l'environnement applicatif GeoNature. A l'heure actuelle, ce sont les responsables des SI du PNE et du PNC qui pilotent GeoNature. Ils sont soutenus techniquement par deux développeurs, l'un s'occupant davantage du backend, et l'autre du frontend. Cette petite équipe travaille quasiment essentiellement sur l'application depuis déjà plusieurs années. Les pilotes des développements sur GeoNature ont d'ailleurs participé à sa création, ce qui leur confère légitimement le rôle de "décideur".

Cependant, les besoins des utilisateurs du PNE sur GeoNature ne sont pas les mêmes que ceux d'autres utilisateurs issus d'autres structures. C'est d'ailleurs pour cela que l'environnement applicatif a été conçu de manière générique et modulable. Mais cette variété des besoins pose des problèmes de gouvernance quant à l'évolution que doit prendre l'application. Par exemple, le SINP PACA a souhaité mettre en place le floutage des données privées et sensibles dans la Synthèse, mais c'est un besoin propre aux SINP régionaux et à l'INPN (niveau national). Ainsi, l'intégration de ces développements dans le cœur de GeoNature n'est pas la priorité de l'équipe de pilotage du PNE, d'autant plus qu'ils doivent à la fois relire et corriger les nombreuses Pull Request (demande d'ajout de développements) des autres développeurs.

Par ailleurs, l'intégration de ce développement ne viendrait pas modifier le fonctionnement de la Synthèse puisqu'il suffirait de désactiver un paramètre de configuration pour que le floutage des données ne soit pas appliqué.

Pour résumer, au plus les développements tarderont à être intégrés, au plus la branche de travail "feat/sinp" divergera par rapport à la branche "develop", au plus l'application Silene Expert sera difficile à maintenir à jour.

Une solution envisageable serait de mettre une personne à temps plein qui travaille à la relecture et la correction des Pull Request sur GeoNature. Ainsi, les développements de chacun seraient plus rapidement intégrés, ce qui optimiserait le temps de travail alloué aux nouveaux développements, ainsi l'application pourrait évoluer plus rapidement.

MIGRATION DES MODULES CONSERVATION DE LA FLORE

GeoNature a été conçu pour fonctionner en briques modulaires. Chaque protocole, répondant à une question scientifique, est amené à avoir son propre module comportant son modèle de base de données (dans un schéma séparé), son API et son interface utilisateur. Les modules développés s'appuient sur le cœur de GeoNature qui est constitué d'un ensemble de briques réutilisables.

Les modules de conservation de la flore sont liés à deux projets où le CBNA est impliqué : le projet Flore Sentinelle pour Suivi Flore Territoire et Suivi Habitat Territoire, et le projet FloreClim pour Flore Prioritaire et Suivi Habitat Station.

- Suivi Flore Territoire vise à étudier la présence d'une espèce définie sur des territoires alpins. Il permet ainsi de suivre l'évolution des espèces dans le temps.

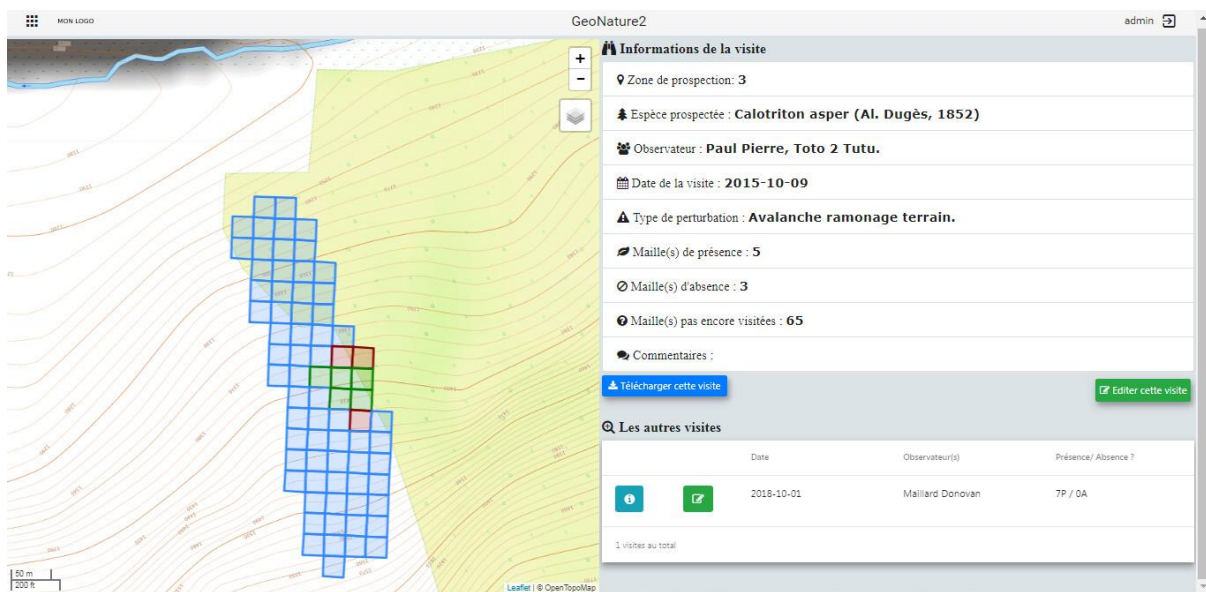


Figure 24 : Interface du module Suivi Flore Territoire, source : GeoNature

- Suivi Habitat Territoire est identique à Suivi Flore Territoire à l'exception qu'il est dédié à l'étude des habitats (échelle de prospection plus grande).

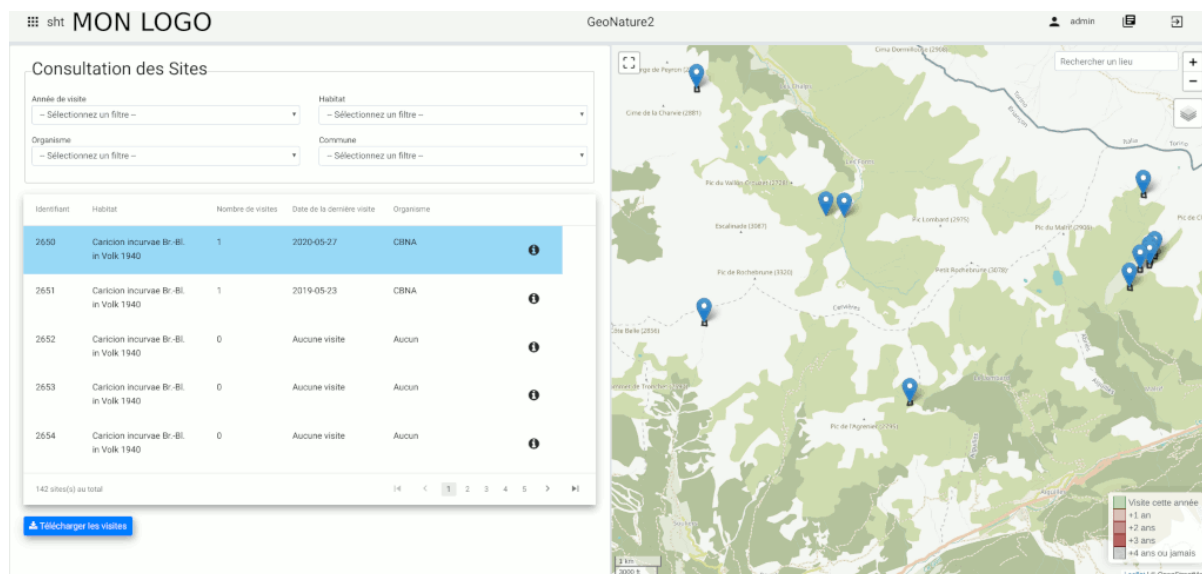


Figure 25 : Interface du module Suivi Habitat Territoire, source : GeoNature

- Flore Prioritaire est un outil de prospection qui agrège toutes les données susceptibles d'aider à la compréhension des évolutions de populations d'espèces patrimoniales ou invasives. Pour ce protocole, il s'agit de délimiter des zones de prospection d'une espèce et de tracer une aire de présence au sein de la zone si celle-ci est observée.

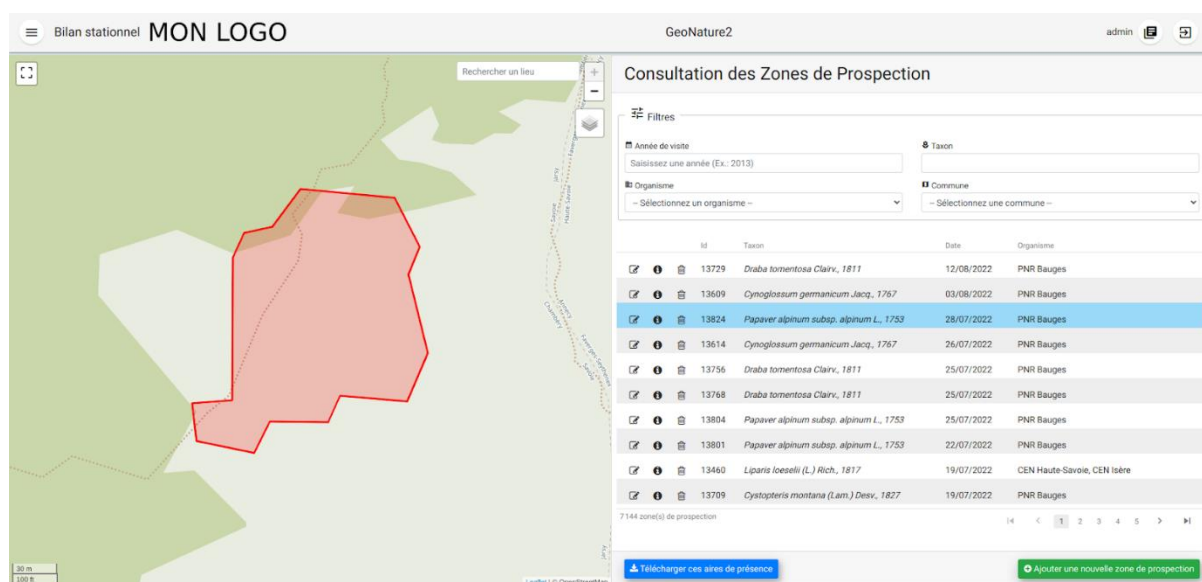


Figure 26 : Interface du module Flore Prioritaire, source : GeoNature

- Suivi Habitat Station vise à comprendre la dynamique des populations et des compositions végétales sur une station donnée (un site).

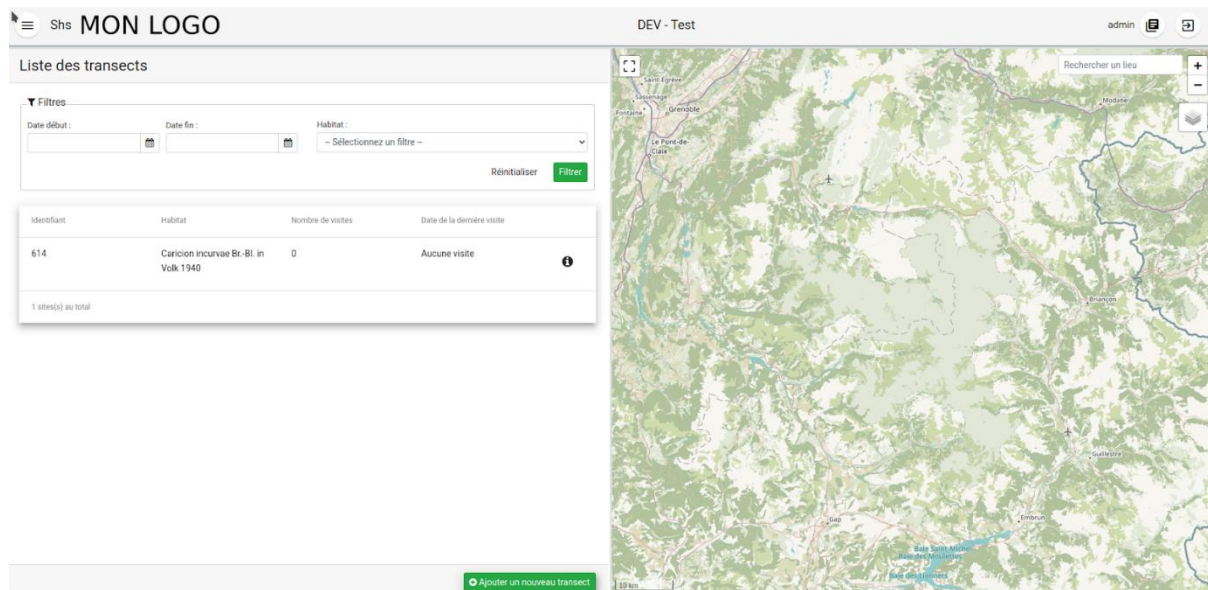


Figure 27 : Interface du module Suivi Habitat Station, source : GeoNature

Il est possible de consulter, rechercher et exporter les données de chaque module. Les droits sont définis de manière centralisée, dans le module partagé UsersHub. Il permet de gérer les droits par utilisateur ou par groupe. C'est le module Admin de GeoNature qui permet de gérer les permissions (Lire, Créer, Modifier, Supprimer, Valider, Exporter), par portée (Mes données, Les données de mon organisme, Toutes les données) et par module.

La mise à jour de ces quatre modules était nécessaire. Au CBNA, le module Flore Prioritaire fonctionnait encore sur une version antérieure de GeoNature, la v1 (voir figure 27). De ce fait, l'interface utilisateur était vieillissante et beaucoup de fonctionnalités pratiques telles que le packaging du module ainsi que l'implémentation d'Alembic n'étaient pas disponibles. De plus, le module devenait difficilement maintenable car les technologies utilisées par GeoNature ont depuis beaucoup évolué. Les trois autres modules quant à eux, bénéficient déjà d'une interface utilisateur plus moderne puisqu'ils fonctionnent sur la version 2.4.1 de GeoNature. Cependant, le packaging python des modules et l'utilisation d'Alembic sont disponibles depuis la version 2.7.5 de GeoNature. Ainsi, le CBNA a souhaité mettre à jour ces quatre modules jusqu'à la dernière version stable disponible (branche "master"), la version 2.9.2.

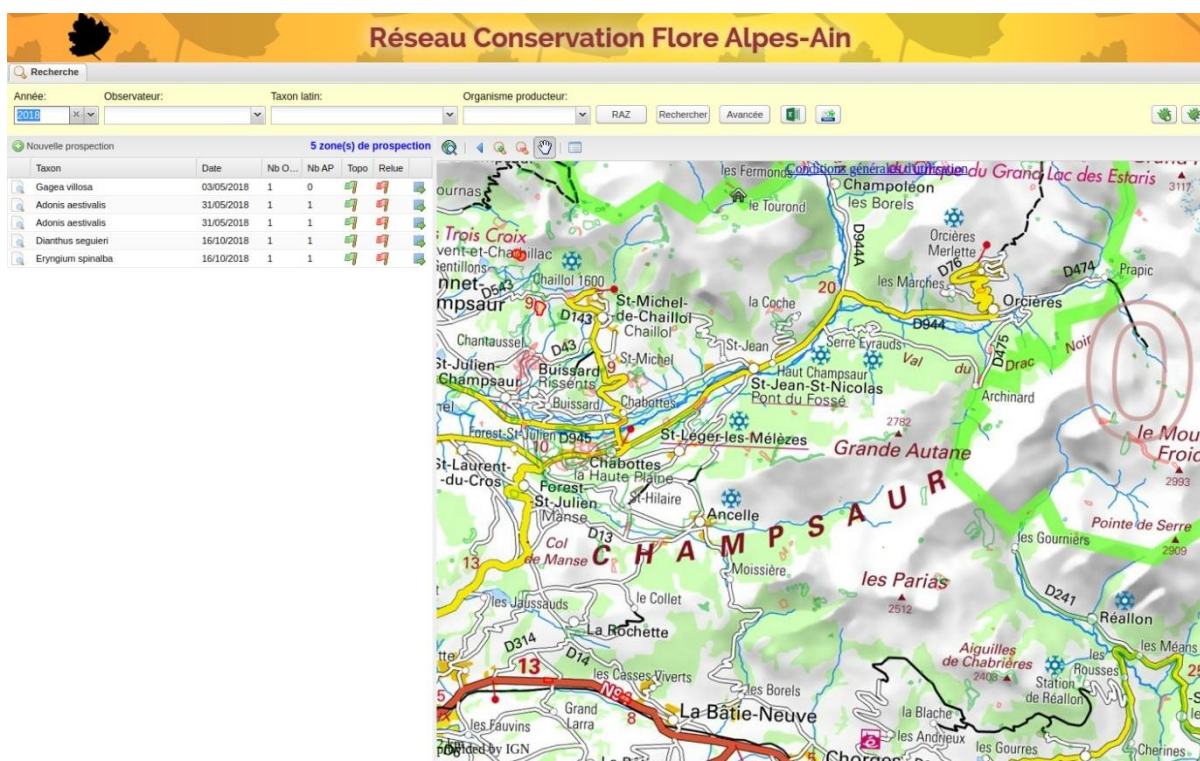


Figure 28 : Ancienne interface du module Flore Prioritaire dans GeoNature v1, source : GeoNature

Par ailleurs, le module Flore Prioritaire, également utilisé par des agents du PNE, a été récemment mis à jour jusqu'à la version 2.8 de GeoNature. Ainsi, nous disposons d'un exemple de module fonctionnel sur une version récente de GeoNature. Les étapes de développement (commits) pour monter en version le module sont disponibles sur le dépôt Git dédié.

MONTÉE EN VERSION DES MODULES

L'objectif de ce développement est de réaliser une montée en version des modules Conservation afin de les adapter à la dernière version stable de GeoNature (2.9.2). Jusqu'à présent, l'installation des modules était effectuée en exécutant des scripts Bash, qui permettaient d'installer le module, les données de référence, la base de données dans l'application GeoNature ainsi que de désinstaller le module. Le but est de faire évoluer ces modules en version "packagé" (voir page suivante) et ainsi se débarrasser des scripts bash. De plus, cela simplifiera l'installation et la désinstallation du module puisqu'il suffira de lancer une commande pip dans l'environnement virtuel. Enfin, ce format packagé combiné aux fichiers de révision Alembic, qui permettent d'indiquer l'état d'une base de données, simplifieront grandement la récupération des modules par d'autres utilisateurs.

Packaging des modules

Le package peut être interprété comme une structure de répertoire de fichiers hiérarchique qui définit un environnement d'application. L'avantage du package Python est de pouvoir utiliser des outils tels que "pip", qui facilite le téléchargement et l'installation du package par d'autres développeurs. De plus, les paquets sont un moyen de structurer les espaces de nommage des modules Python en utilisant une notation "pointée". Par exemple, le nom de module "A.B" désigne le sous-module B du paquet A.

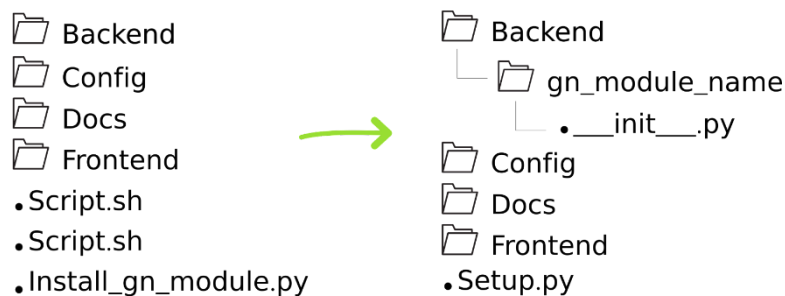


Figure 29 : Modification des répertoires et fichiers de travail pour le packaging des modules, source : Nicolas Rouyer

Le dossier `gn_module_name` comporte les fichiers python de création des routes Flask, du modèle de données SQLAlchemy ainsi que la configuration du module, essentiels à sa création.

Création des révisions 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. Cela signifie que la suite d'opérations de la transaction est indivisible, en cas d'échec en cours d'une des opérations, la suite d'opérations doit être complètement annulée (*rollback*) quel que soit le nombre d'opérations déjà réussies.

Les opérations contenues dans les fichiers de migration Alembic sont enregistrés dans la table `public.alembic_version` ce qui permet à Alembic de savoir dans quel état la base de données se trouve.

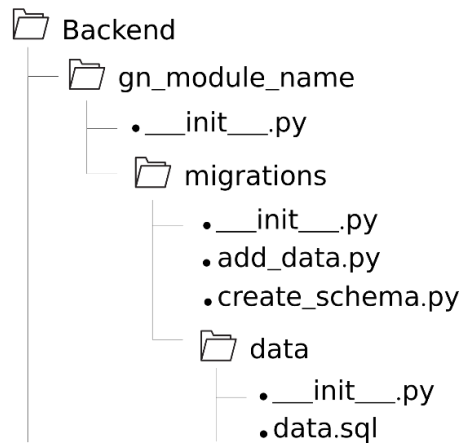


Figure 30 : Répertoires et fichiers de travail des modules packagés

Les fichiers de migrations du module se trouvent dans le dossier `backend/gn_module_name/migrations/`. Pour que ceux-ci soient détectés par Alembic, il suffit de définir un point d'entrée dans le `setup.py` à la racine du module :

```

setuptools.setup(

    ...,
    entry_points={
        'alembic': [
            'migrations = gn_module_name:migrations',
        ],
    },
    ...
)

```

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, chaque module dispose de sa propre branche.
- 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-dit 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 les branches de chaque module reposent sur l'installation préalable du module des nomenclatures communes.

Nous avons créé et adapté pour chacun des modules deux fichiers de révision Alembic :

- Le premier "add_data" permet d'ajouter dans la BDD les données de référence nécessaires au fonctionnement du module et à la création du schéma dans la BDD.
- Le second "create schema" dépend du fichier précédent, il permet de créer le schéma dans la base GeoNature ainsi que les tables et les différentes relations.

Le dossier *data* contient l'ensemble des fichiers SQL nécessaires à l'insertion des données dans la base de données de GeoNature.

Enfin, les modules Conservation fonctionnant avec le même type de données, un cinquième module²⁹ a été créé, faisant office de bibliothèque partagée pour les données communes entre ces modules. Ce module est composé essentiellement de fichiers de révision Alembic. Les quatre modules Conservation reposent sur l'installation préalable de ce module partagé.

MIGRATION VERS LE NOUVEAU SERVEUR

Une fois les quatre modules fonctionnels sur GeoNature 2.9.2, et le nouveau serveur préparé, nous (les deux stagiaires et le développeur) effectuerons leur migration fin septembre. Les étapes de la migration s'organisent comme suit :

1. Les agents du CBNA sont prévenus par mail que le serveur sera en maintenance pendant une journée.
2. L'accès à l'ancien serveur est bloqué
3. La base de données GeoNature de Flore Sentinelle est récupérée sur le nouveau serveur (PostgreSQL et l'extension PostGIS ont préalablement été installés)
4. Récupération de l'installation GeoNature sur le nouveau serveur (v 2.4.1)
5. Mise à jour de GN vers 2.9.2 en suivant les releases (notes de montée en version)
6. Installation des 4 modules packagés
7. Vérification du bon fonctionnement global
8. Réouverture des applications sur le nouveau serveur

Ces étapes seront réalisées en suivant la documentation technique rédigée durant le stage, ce qui nous permettra d'effectuer la migration sans difficulté et en un minimum de temps.

²⁹ https://github.com/cbn-alpin/gn_conservation_backend_shared

BILAN

Résultats et pistes d'amélioration

Le travail effectué lors de ce stage a permis de faire évoluer les outils métier des agents du CBNA. Le travail sur la Synthèse de GeoNature a été intégré dans Silene Expert, les utilisateurs disposent donc d'un nouveau mode d'affichage des données sur la Synthèse et les performances de l'API sont désormais optimisées grâce au travail sur l'agrégation des données par géométrie. Cependant à ce jour, ces développements n'ont pas encore été intégrés à la branche "develop" de GeoNature.

Les membres du service Conservation pourront bénéficier de quatre modules évolués et fonctionnels. Leur montée en version permet également à d'autres structures de pouvoir les installer facilement. Les modules peuvent encore être améliorés côté backend et côté frontend. C'est d'ailleurs l'étape qui suivra la migration des modules sur le nouveau serveur. Un échange avec les utilisateurs sera réalisé pour cibler leurs besoins spécifiques. Enfin, des triggers (déclencheurs d'action) seront créés dans la base de données afin d'insérer et mettre à jour de manière automatique les données de chaque module vers la Synthèse de GeoNature. Ainsi, les utilisateurs de Silene Expert et de Biodiv'AURA Expert auront accès aux données produites par le CBNA sans avoir besoin de passer par des scripts bash d'intégration des données.

Enfin, la documentation technique rédigée au cours du stage sur le DokuWiki du CBNA vient étoffer les aides au développement déjà existantes.

Montée en compétences

L'ordre dans lequel les missions de ce stage ont été effectuées m'ont permis de faire la transition de mes compétences en géomatique vers des compétences en développement web. Cela a également été possible grâce à un encadrement et un accompagnement de qualité de la part du développeur du CBNA.

Globalement, je suis fortement montée en compétence en informatique. En effet, travailler sur un environnement applicatif aussi riche en technologies que GeoNature m'a permis à la fois de m'initier au fonctionnement du framework Angular avec TypeScript côté frontend, et aux frameworks Python SQLAlchemy et Alembic côté backend. Ce stage m'a également permis de renforcer mes compétences en HTML, Javascript, SQL et Python ainsi que son framework Flask. Développer au quotidien sur ces différents langages informatiques m'a permis de mieux comprendre les concepts de développement web, notamment le fonctionnement d'une API web REST.

Évoluer sur un outil collaboratif m'a fait fortement monter en compétences sur l'utilisation de Git et Github. De plus, travailler sur un PC Linux m'a permis d'utiliser au quotidien le terminal, j'ai ainsi pu me familiariser avec beaucoup de commandes Psq, Git et Bash.

Afin de pouvoir effectuer les missions de stage, j'ai dû rapidement monter en compétences sur tous ces aspects ce qui a renforcé ma capacité à m'adapter à un nouvel environnement de travail. De plus, développer conjointement avec deux autres personnes a renforcé mes aptitudes à travailler en équipe.

Enfin, travailler au sein d'un Conservatoire botanique m'a permis d'en apprendre davantage sur les métiers d'écologues et de botanistes, et sur la flore alpine en général.

Perspectives

Ce stage concluant mes deux années de reconversion professionnelle a été une expérience très riche, surtout par rapport à la montée en compétences techniques. Ce stage m'a permis d'affiner mon orientation professionnelle. Participer au développement de l'environnement applicatif GeoNature et découvrir les métiers liés à la préservation du patrimoine naturel confirme mon intention de travailler dans le domaine de l'Environnement. Je souhaite également continuer à progresser sur le volet informatique de la géomatique. Enfin, j'ai apprécié évoluer au sein de l'équipe du CBNA.

Le CBNA m'a proposé un contrat de deux ans pour continuer à travailler sur GeoNature, aider le développeur à administrer les outils métiers des SINP PACA et AURA ainsi que de maintenir les modules de Conservation de la flore et développer de nouveaux modules liés aux protocoles scientifiques du CBNA. J'ai accepté cette opportunité qui me permettra de continuer à monter en compétences en développement web et de continuer à utiliser mes compétences en géomatique en travaillant sur des outils web cartographiques. Je devrai également monter en compétences en administration des serveurs et ainsi disposer de compétences solides en informatique.

Je remercie l'équipe pédagogique du Master Géographies Numériques qui m'a permis de trouver aujourd'hui un poste qui me plaît et qui me permet d'évoluer encore davantage professionnellement.

Bibliographie/Webographie

- R. C. Martin et H. Soulard, *Coder proprement*. Paris: Pearson, 2009.
- *Deviens un ninja avec Angular2*, Ninja Squad.
- T. Fredrich, *RESTful Service Best Practices*.

<http://www.cbn-alpin-biblio.fr/Record.htm?idlist=1&record=19646473124914646559>

<http://cbnfc-ori.org/cbnfc-ori/le-reseau-des-conservatoires-botaniques-nationaux>

Index des figures

Figure 1 : Territoire d'agrément du CBNA, source: CBNA	6
Figure 1 : Territoire d'agrément du CBNA, source: CBNA	6
Figure 2 : Les données du CBNA, source : CBNA	10
Figure 3 : Les serveurs de données du CBNA, source : Nicolas Rouyer	11
Figure 4 : Fonctionnement de GeoNature, source : documentations officielle GeoNature	13
Figure 5 : Les technologies de GeoNature, source : documentation officielle de GeoNature	13
Figure 6 : Fonctionnement de l'environnement applicatif GeoNature au CBNA, source : Cendrine HOARAU	14
Figure 7 : Cheminement des données du CBNA depuis le terrain jusqu'à la plateforme nationale de l'INPN, source : Cendrine HOARAU	15
Figure 8 : Le module Synthèse de GeoNature Expert, source : GeoNature	16
Figure 9 : Fonctionnement des dépôts Github, source : Cendrine HOARAU	17
Figure 10 : Spirale de markers Leaflet, source : Leaflet	19
Figure 11 : MarkerCluster Leaflet, source : Leaflet	19
Figure 12 : Ancien format de récupération des données de la Synthèse, source : Cendrine Hoarau	20
Figure 13 : Nouveau format de récupération des données dans la Synthèse, source : Cendrine HOARAU	20
Figure 14 : Proposition de style pour les CircleMarker, source : Leaflet	22
Figure 15 : Style des CircleMarker utilisé dans la Synthèse, source : Leaflet	22
Figure 16 : Affichage par maille dans la Synthèse GeoNature, source : GeoNature	23

<i>Figure 17 : Affichage par points dans la Synthèse GeoNature avec le floutage des données,</i> <i>source : GeoNature</i>	25
<i>Figure 18 : Affichage par mailles dans la Synthèse GeoNature avec le floutage des données,</i> <i>source : GeoNature</i>	27
<i>Figure 19 : Répertoires de travail côté frontend pour la Synthèse GeoNature, source :</i> <i>Cendrine Hoarau</i>	27
<i>Figure 200 : Zoom sur une pop-up au clic sur une maille dans la Synthèse GeoNature,</i> <i>source : GeoNature</i>	29
<i>Figure 21 : Légende associée aux mailles sur la Synthèse GeoNature, source : GeoNature</i>	30
<i>Figure 22 : Observations sélectionnées dans la liste des observations de la Synthèse, source :</i> <i>GeoNature</i>	30
<i>Figure 23 : Interface du module Suivi Flore Territoire, source : GeoNature</i>	33
<i>Figure 24 : Interface du module Suivi Habitat Territoire, source : GeoNature</i>	34
<i>Figure 25 : Interface du module Flore Prioritaire, source : GeoNature</i>	34
<i>Figure 26 : Interface du module Suivi Habitat Station, source : GeoNature</i>	35
<i>Figure 27 : Ancienne interface du module Flore Prioritaire dans GeoNature v1, source :</i> <i>GeoNature</i>	36
<i>Figure 28 : Modification des répertoires et fichiers de travail pour le packaging des modules,</i> <i>source : Nicolas Rouyer</i>	37
<i>Figure 29 : Répertoires et fichiers de travail des modules packagés</i>	38

Annexes

Annexe 1 : Requête SQL d'agrégation des données par géométrie

```

with observations as (
  select
    s.the_geom_4326,
    s.id_synthese,
    s.unique_id_sinp::varchar as unique_id_sinp,
    s.cd_nom as cd_nom,
    s.lb_nom as lb_nom,
    COALESCE(s.nom_vern, s.lb_nom) as nom_vern_or_lb_nom,
    CASE
      WHEN s.count_min != s.count_max THEN s.count_min || ' - ' ||
s.count_max
      WHEN s.count_min IS NOT NULL THEN s.count_min::varchar
      ELSE "
    end as count_min_max,
    s.observers,
    s.dataset_name,
    s.date_min::varchar,
    s.url_source
  FROM
    gn_synthese.v_synthese_for_web_app s
  where
    s.the_geom_4326 is not null
  order by
    s.date_min desc
  limit 10000
)
SELECT
  st_asgeojson(s.the_geom_4326) AS geometry,
  json_agg(
    json_build_object(
      'id', s.id_synthese,
      'unique_id_sinp', s.unique_id_sinp,
      'cd_nom', s.cd_nom,
      'lb_nom', s.lb_nom,
      'nom_vern_or_lb_nom', s.nom_vern_or_lb_nom,
      'count_min_max', s.count_min_max,
      'observers', s.observers,
      'dataset_name', s.dataset_name,
      'date_min', s.date_min,
      'url_source', s.url_source
    )
  ) AS properties
from observations as s
GROUP BY
  the_geom_4326

```


Annexe 2 : Extraits de code du fichier synthse-carte.component.ts

```

if (geojson.geometry.type == 'Polygon' ||
    geojson.geometry.type == 'MultiPolygon') {
    const latLng = L.GeoJSON.coordsToLatLngs(
        geojson.geometry.coordinates,
        geojson.geometry.type === 'Polygon' ? 1 : 2
    );
    if (this.areasEnable) {
        this.setAreasStyle(new L.Polygon(latLng),
            geojson.properties.observations.id);
    } else {
        this.setStyleEventAndAdd(
            new L.Polygon(latLng).bindTooltip(`${countObs}
            observation(s)`),
            geojson.properties.observations.id
        );
    }
}

private setAreasStyle(layer, ids) {
    this.originAreasStyle['fillColor'] =
    this.getColor(ids.length);
    layer.setStyle(this.originAreasStyle);
    this.eventOnEachFeature(ids, layer);
    this.cluserOrSimpleFeatureGroup.addLayer(layer);
}

private getColor(obsNbr) {
    let classesNbr =
    this.config['SYNTHESE']['AREA_AGGREGATION_LEGEND_CLASSES'].length;
    let lastIndex = classesNbr - 1;
    for (let i = 0; i < classesNbr; i++) {
        let legendClass =
    this.config['SYNTHESE']['AREA_AGGREGATION_LEGEND_CLASSES'][i];
        if (i !== lastIndex) {
            if (obsNbr > legendClass.min) {
                return legendClass.color;
            }
        } else {
            return legendClass.color;
        }
    }
}

```