



## Cahier des charges

# Développement du module d'imports de GeoNature

Date limite de réponse : **mercredi 6 mars 2019 à 17h**



**La Région**   
Auvergne-Rhône-Alpes



## Commanditaire



**Flavia Association pour les Papillons et leur Étude**  
10 Route de Cozance  
38460 Trept



**La Région**   
Auvergne-Rhône-Alpes

Dans le cadre d'un projet financé par la **DREAL Auvergne-Rhône-Alpes** et le **Conseil Régional Auvergne-Rhône-Alpes**.

## Sommaire

|  |    |
|--|----|
| Commanditaire  | 2  |
| Sommaire   | 2  |
| Contexte général du projet                           | 3  |
| Objet de la mission                                  | 4  |
| Description détaillée de la mission                  | 4  |
| Fonctionnement global attendu du module              | 4  |
| Mécanisme et étapes du processus                     | 5  |
| Options  | 11 |
| Modalité de mise en oeuvre                           | 12 |
| Livrables et modalités de livraison                  | 13 |
| Calendrier   | 14 |
| Contacts   | 14 |
| Modalités de réponse                                 | 14 |
| Annexe 1 : Schéma du mécanisme et étapes de l'import | 15 |
| Annexe 2 : Ebauche de MCD pour le module             | 16 |

## Contexte général du projet

Flavia - Association pour les Papillons et leur Étude - est une association loi 1901 créée il y a 20 ans et spécialisée dans l'expertise entomologique (<http://www.flavia-ape.fr>). L'objectif de cette structure est d'étudier les papillons dans une optique de conservation des espèces et de leurs milieux, dans les Alpes et leur périphérie.

Depuis début 2018, Flavia APE est également porteur du projet de Pôle Invertébrés d'Auvergne-Rhône-Alpes, actuellement en préfiguration (<https://www.pole-invertebres.fr>). Ce pôle a pour vocation de fédérer le réseau des spécialistes travaillant sur la faune invertébrée en région, et de centraliser et rediffuser les données naturalistes produites par chacun afin de mutualiser la connaissance. Le projet s'inscrit dans le cadre du SINP (<http://www.naturefrance.fr/sinp/presentation-du-sinp>), et est piloté conjointement par Flavia APE, la DREAL Auvergne-Rhône-Alpes et le Conseil Régional d'Auvergne-Rhône-Alpes. Afin de mener à bien ses missions, le Pôle Invertébrés en préfiguration s'appuie notamment sur la suite d'outils GeoNature.

Les outils GeoNature sont un ensemble d'applications open-source permettant de saisir, gérer, synthétiser et diffuser les données faune et flore d'un territoire (<http://geonature.fr>).

Ils permettent ainsi de déployer un système d'informations complet pour la gestion des données Faune/Flore, allant de :

- la gestion des référentiels (taxonomiques, utilisateurs, géographiques et nomenclatureaux),
- à la saisie web et mobile dans différents protocoles,
- à la gestion de leurs métadonnées,
- à l'intégration de données de partenaires,
- à l'export des données selon les formats attendus par chaque partenaires
- à la synthèse des données des différents protocoles sous forme de DEE
- à la validation scientifique des données
- à la diffusion des données sur un portail web grand public

Les applications de la suite GeoNature sont publiées sous licence libre et sont maintenues par un collectif de développeurs au sein des Parcs Nationaux (<https://github.com/PnX-SI/GeoNature>). Ces outils sont de plus en plus reconnus dans le milieu naturaliste et sont désormais utilisés par un nombre croissant d'acteurs (Parcs nationaux, plusieurs PNR, Flavia APE, Conservatoires etc). Chaque structure dispose de sa propre instance.

La suite applicative est en fin de refonte vers sa version 2, et se compose de :

- GeoNature : application au coeur du système, assurant la gestion des données naturalistes
- TaxHub : application de gestion de la taxonomie : informations sur les espèces, regroupements d'espèces, médias etc
- UsersHub : application de création et gestion des droits d'utilisateurs
- GeoNature-atlas : application de diffusion des données naturalistes synthétisées à destination du grand public
- GeoNature-mobile : application mobile de saisie des données naturalistes (en développement)
- GeoNature-Citizen : application de recensement des observations dans le cadre de sciences participatives (en développement)

L'outil GeoNature, au coeur de l'ensemble, se compose lui-même :

- d'un coeur, avec un module de saisie (occtax), un module de consultation (synthese), un module d'administration et un module de métadonnées,
- de modules génériques tels qu'un module d'export des données et un module de validation. C'est dans ce cadre que s'inscrira le module développé lors de cette mission.
- et de modules spécifiques, pouvant être développés et déployés par les différents utilisateurs pour répondre à leurs besoins précis : saisies de données d'un protocole scientifique particulier etc

La synthèse consiste alors à tirer de chaque module les informations qui y sont communes (« qui a vu quoi, où, quand et comment ? »), et les centraliser sous un format homogène.

## Objet de la mission

Si la base de données PostgreSQL sur laquelle s'appuie GeoNature permet bien l'import et la centralisation de données partenaires, un certain nombre d'éléments rendent la tâche chronophage et technique pour son administrateur. Ce travail, central pour l'activité du Pôle Invertébrés en préfiguration et d'autres utilisateurs de GeoNature, doit être optimisé.

Cette prestation a pour but le développement d'un module de GeoNature permettant de faciliter l'import des données naturalistes provenant de différents formats, et leur « traduction » vers le format de synthèse à l'aide d'une interface web.

Le module développé s'adressera en priorité aux administrateurs des différentes instances de GeoNature, en charge de la gestion de la base de données, ou à des utilisateurs avertis.

## Description détaillée de la mission

Le module d'import aura pour objectif de permettre aux utilisateurs ayant les droits nécessaires d'importer des lots de données dans la base GeoNature. Les données importées auront un format qui leur est propre mais devront être traduites dans un format standard.

Le module devra prendre en charge les données issues de fichiers shapefile, csv et json. Il pourra s'appuyer en partie sur les fonctions d'imports implémentées dans la base Postgresql de GeoNature (<https://geonature.readthedocs.io/fr/latest/admin-manual.html#fonctions>)

L'outil devra donc permettre l'upload de fichier, leur lecture et leur mise en conformité avec les formats de données attendus dans la table "synthese" de l'outil.

### Fonctionnement global attendu du module

Le module procèdera à l'import de données depuis des fichiers sources de structures variables, vers une table d'imports comprenant tous les champs sources, et pouvant être enrichie.

Au cours de l'import, le module enrichira la table d'imports avec deux champs *gn\_is\_valid* (boolean) et *id\_import*. L'administrateur de la base pourra également l'enrichir manuellement selon ses besoins (ajout d'un champ *gn\_cd\_nom*, compléments sur des données incomplètes, jointures sur des géométries par exemple).

*In fine* le module utilisera cette table d'import enrichie pour alimenter la synthèse de GeoNature.

**Dans son fonctionnement, le module devra enregistrer en direct dans la base de données les informations saisies par l'utilisateur. Inversement, les formulaires de l'interface devront tenir compte en direct (ou avec un bouton "actualiser") des modifications apportées dans la base de données (structure et contenus).**

Ces informations resteront durablement dans la base de données, permettant :

- de stopper et reprendre en tout temps un travail d'import commencé
- de réutiliser dans un nouvel import des mappings déjà effectués par le passé (format source identique à un import précédent)

Au fur et à mesure de l'import, le module pourra identifier des erreurs ou des lacunes (informations manquantes, données de type invalide etc). Dans ce cas, il devra notifier l'utilisateur en interface ("*XX données ne sont pas valides pour le format attendu*") et renseigner le champ *gn\_is\_valid* à "false" en base. Tant que la table d'imports contiendra des données "*gn\_is\_valid=f*", l'import restera "en cours". Lorsque toutes les données seront valides, l'utilisateur pourra terminer l'import : les données seront poussées dans la table synthèse et l'import sera terminé.

Le module ne permet pas de revenir sur des imports terminés, ni sur les données précédemment importées jusqu'à la synthèse.

A la fin de la manipulation, s'il reste des erreurs, l'utilisateur pourra au choix :

- laisser son import en cours (aucune données n'est alors poussée dans la synthèse) et faire les corrections en base avant de poursuivre
- exporter les données invalides (*is\_valid=f*) dans son format source et les supprimer de la table d'import, permettant de finaliser l'import et pousser les données dans la synthèse (L'utilisateur pourra faire un nouvel import de ces données invalides après correction sur son fichier source s'il le souhaite).

A tout moment, un utilisateur doit également pouvoir **annuler** son import, il perd alors toutes les informations qu'il avait saisies : mappings et tables sont retirées de la base, le fichier source est retiré du serveur.

## Mécanisme et étapes du processus

voir schéma - Annexe 1

---

### Etape 1 : Connexion - visualisation des derniers imports effectués

GeoNature repose sur un ensemble de droits (CRUVED) et de portées (0 à 3), permettant de définir pour chaque module les actions autorisées à chaque rôle (personnes ou groupes).



La gestion des droits par le CRUVED

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

Le module développé devra permettre de visualiser l'historique des imports effectués aux utilisateurs disposant des droits nécessaires (R>0), comprenant quelques informations simples : date de l'import dans la synthèse, volume de données, organisme, nombre de taxons... (éventuellement paramétrable : cf configuration d'occtax et de la synthèse).

Pour cela, chaque import sera historisé et associé à ses informations globales dans la base de données (annexe 2 : *t\_imports*).

Outre cet historique, les imports en cours seront également consultables avec quelques informations simples (date de début de l'import, organisme, jeu de données).

Les étapes suivantes ne seront accessibles qu'avec des droits d'écriture (C).

---

## Etape 2 : Choix de l'action

Les utilisateurs disposant de droits d'écriture (C>0) auront alors le choix entre sélectionner l'un de ces imports en cours pour le **poursuivre**, ou **créer un nouvel import**.

1 - dans le cas d'un nouvel import, l'utilisateur est amené à l'étape 3 pour uploader ses données.

2 - dans le cas de la poursuite d'un import, le module renvoie aux étapes suivantes mais les formulaires suivants seront grisés (upload) ou pré-remplis (mais modifiables : SRID, encodage, mappings) jusqu'au stade où le travail avait été effectué précédemment. Le jeu de données doit être actif, et l'utilisateur doit avoir des droits d'écriture suffisants pour poursuivre l'import des données.

---

## Etape 3 : Sélection du jeu de données

Dans GeoNature, les données sont organisées au sein de jeux de données (eux-même organisés au sein de cadre d'acquisitions et rattachés à des organismes).

**Dans le cadre du module d'import, nous définissons qu'un jeu de données correspond à une table d'imports spécifique, et donc à un format et un SRID uniques.**

L'utilisateur doit sélectionner un jeu de données actif créé en amont et sur lesquels il a les droits :

- si aucun import n'a été effectué précédemment pour le jeu de données sélectionné par l'utilisateur, le format source n'est pas connu : aucune contrainte n'est imposée à ce stade.

- si un import a déjà été effectué auparavant pour le jeu de données sélectionné, la table d'import aura déjà été créée par le passé. Sa structure contraindra le format source attendu (en ignorant les champs enrichis) et le système de projection (SRID).

Le format de données source attendu peut alors être téléchargé (.csv) par l'utilisateur.

---

## Etape 4 : Upload du fichier source

Le module devra permettre d'uploader 3 formats de fichiers (et refuser tout autre format) :

- fichier tableur au format csv avec en-tête
- fichiers shapefile (et ses dépendances, dont la présence est à contrôler) au format zip
- fichier json

L'utilisateur devra alors en préciser l'encodage.

S'il s'agit d'un premier import pour le jeu de données sélectionné, le SRID doit être défini par l'utilisateur pour les fichiers csv.



Le répertoire de stockage des fichiers ainsi que la taille maximale des fichiers acceptés devront être paramétrables dans la configuration du module.

## Etape 5 : Import des données sources et archivage

L'import des données sources peut se faire selon deux cas :

### Cas 1 : Il s'agit du premier import pour le jeu de données sélectionné

Si l'utilisateur procède à son premier import pour le jeu de données sélectionné, alors **une table d'import est créée** sur le modèle de la table source. Tous les champs sont conservés ainsi que leurs entêtes. Tous ces champs sont de type *text*.

A ce stade, aucune clé primaire n'est encore déclarée (elle peut ne pas exister dans le fichier source), mais une clé sera indispensable sur cette table (qui pourra être enrichie) pour les étapes suivantes. Toutes les lignes sont importées.

#### Table d'archive

Une fois le fichier uploadé et sa conformité vérifiée (format, encodage, SRID valide), le module devra créer une table d'archive brute dans un schéma dédié (tous les champs de type *text*) et y importer toutes les données sources. Cette archive ne sera plus réutilisée par le module et n'a pas vocation à être modifiée à terme : elle ne présente donc aucune contrainte (clé primaire etc).

Cette étape peut s'appuyer en partie sur la commande "*gn\_imports.load\_csv\_file*" déjà implémentée dans l'outil pour les fichiers csv.

### Cas 2 : Des imports ont déjà été effectués pour le jeu de données sélectionné

Si l'utilisateur a sélectionné un jeu de données pour lequel des imports ont déjà été effectués par le passé, une table d'import est déjà présente dans la base de données. Celle-ci définit la structure attendue (champs et en-tête), en ignorant les champs enrichis s'ils sont absents de la source, hors clé primaire.

Dans ce cas précis, la clé primaire a obligatoirement été créée dans la table d'imports et elle **doit** donc être présente dans le fichier source uploadé.

Si la structure du fichier uploadé est conforme à celle de la table d'import, cette dernière **est complétée (UPDATE) avec les nouvelles lignes**, sur la base des clés primaires.

#### Les cas particuliers non gérés

Si un fichier source comprend toutes les données d'un jeu de données, dont une partie a déjà été intégrée lors d'un précédent import :

- Seules les données ayant un identifiant primaire absent de la table d'import sont importées
- La mise à jour de données (dans la base) qui auraient été modifiées dans le fichier source depuis un précédent import n'est pas traitée par le module
- La suppression de données (dans la base) qui auraient été supprimées du fichier source depuis un précédent import n'est pas traitée par le module
- Par conséquent, si une donnée avait été supprimée de la base, mais est toujours présente dans le fichier source, elle est réimportée

Ces cas particuliers doivent être gérés par l'administrateur en complément des actions du module. Ces points seront rappelés dans la documentation.

## Fonctionnement des mappings

Le module devra conserver les données sources complètes (éventuellement enrichies) dans la table d'import, mais devra également "traduire" ces données dans le format homogène de la table synthèse.

Pour ce faire, il devra permettre le "mapping" (mise en correspondance) des champs sources de la table d'import vers les champs cibles de la table synthèse (étape 6), puis du vocabulaire source vers les nomenclatures cibles (étape 7).

Afin de mémoriser ce travail, le module enregistrera dans la base l'ensemble des correspondances renseignées dans un mapping (voir annexe 2 : t\_mapping). Chaque mapping (ensemble de correspondances) aura un cd\_mapping et un nom (bib\_mapping). Ce fonctionnement permettra de reprendre un mapping en cours, ou d'utiliser un même mapping pour plusieurs imports.

### Etape 6 : Mapping des champs

Pour que les données de la table d'import puissent alimenter la table synthèse, un premier mapping (mise en correspondance des champs) est nécessaire. Certains champs de la synthèse sont obligatoires (id source, observateurs, date, etc), d'autres peuvent être renseignés à l'aide de valeurs par défaut, et d'autres enfin peuvent ne pas être renseignés (UUID par exemple).

#### Choix du mapping :

Au début de cette phase, l'utilisateur doit choisir entre **utiliser un mapping déjà existant** (créé lors d'un précédent import), ou **créer un nouveau mapping** (qu'il peut nommer).

S'il utilise un mapping déjà existant, le formulaire de mapping des champs se remplit sur la base des informations contenues dans la base de données. L'utilisateur peut modifier ou compléter ce mapping (les modifications ou compléments se répercutent dans la base de données (annexe 2 : t\_mappings) mais n'ont aucun impact sur les données importées précédemment. La cohérence des modifications et leurs incidences sont à la charge de l'utilisateur).

#### Mapping et interface :

Pour effectuer ce mapping, l'interface devra lister dynamiquement l'ensemble des champs de la table synthèse et, pour chacun, permettre à l'utilisateur de sélectionner l'un de ses champs sources (menu déroulant par exemple). Ce processus pourra s'appuyer pour partie sur la commande "`gn_imports.fct_generate_matching`" implémentée dans l'outil.

#### Transtypage et alertes de non-conformité :

Au cours de ce mapping, les données sources devront se conformer au type attendu dans le champ cible (transtypage). En cas de données non conformes, une alerte devra être faite à l'utilisateur, qui pourra alors visualiser les données en question et :

- modifier son mapping
- taguer les données "invalides" (`gn_is_valid=f`) en vue d'une correction ou suppression ultérieure (étape 9).

#### Champs obligatoires

Les champs obligatoires devront être identifiés et apparents en priorité sur l'interface.

Leur absence dans les données sources empêchera l'utilisateur de poursuivre son travail. Il pourra alors **mettre son import en standby** pour y revenir ultérieurement (après correction de la table d'import dans la bdd), ou **abandonner** son import (la table d'archive et la table d'import sont alors supprimées, ainsi que le fichier uploadé sur le serveur et les informations de mapping).

Une fois le mapping des champs effectué et la conformité des types vérifiée, l'utilisateur pourra prévisualiser le résultat afin de corriger son mapping ou de poursuivre jusqu'à l'étape suivante.

Ce mapping pourra être automatisé autant que possible, sur la base des en-têtes ou des formats de données notamment afin de pré-remplir les correspondances.



## Dates et heures

L'utilisateur pourra définir, en plus du ou des champs correspondant aux dates (avec heures), quel est leur format. Les dates et heures déconcaténées (une colonne jour, une colonne mois, une colonne année...) pourront ne pas être traitées (enrichissement en base si nécessaire).

### Aide au survol

L'intitulé de certains champs de la synthèse n'est pas toujours familier aux utilisateurs.

Une aide au survol devra permettre de donner la description du champ dans les étapes de mapping pour préciser à l'utilisateur l'information attendue.

Ces descriptions sont disponibles dans la base pour les champs relevant de nomenclatures.

## Géométries

Selon les données sources, soit un champ de type *geometry* sera disponible, soit la géométrie devra être calculée sur la base d'un champ X et un champ Y (SRID défini à l'étape 4).

## UUID

L'UUID est l'identifiant unique utilisé dans le cadre du SINP. Selon l'utilisateur et les jeux de données, il sera nécessaire ou non d'en produire un.

Si les données sources comportent un UUID, il doit être conservé grâce au mapping.

Si les données sources ne comportent pas d'UUID (ou si seulement certaines données en ont), l'utilisateur doit pouvoir choisir de créer les UUID manquants ou de laisser le champ vide (case à cocher).

---

## Etape 7 : Mapping de contenu

GeoNature s'appuie sur un ensemble de nomenclatures utilisées au niveau national, dans le cadre du SINP. Certains champs de la synthèse n'acceptent que les identifiants de ces nomenclatures, permettant d'homogénéiser le vocabulaire des données consultables et facilitant les analyses à posteriori. Il est alors nécessaire d'établir une correspondance entre le vocabulaire des données sources et les nomenclatures attendues dans la table cible.

Après avoir effectué le mapping des champs, il est possible d'identifier l'ensemble du vocabulaire utilisé dans le champ source, et l'ensemble des possibilités attendues dans le champ cible. L'interface devra lister les modalités existantes dans le champ source, et permettre de choisir pour chaque modalité la valeur cible souhaitée. Plusieurs valeurs du champ source peuvent avoir une même valeur cible.

Si un mapping existant a été sélectionné à l'étape 6, le formulaire devra être pré-rempli sur la base des informations stockées dans la base de données, et complété si de nouvelles valeurs sont présentes dans les données sources.

Les valeurs cibles par défaut sont déjà disponibles dans la base de données, dans la table `gn_synthese.defaults_nomenclatures_value` (voir la fonction "`get_default_nomenclature_value`").

Ce mapping doit être effectué pour chacun des champs sources correspondant à une nomenclature de la synthèse.

De la même manière qu'à l'étape 6, une aide au survol est attendue sur la base des données disponibles dans la base de données.

---

## Etape 8 : Prévisualisation, contrôle des mappings

A l'issue de ces étapes préparatoires, le module devra permettre à l'utilisateur de prévisualiser ses données (valides) au format de synthèse pour s'assurer que les mappings sont bien conformes à ses attentes avant intégration des données.


Cette prévisualisation se fera sans que les valeurs des champs ne soient codifiées avec les id des nomenclatures (jointures).

---

## Etape 9 : Validation, ajout des données à la synthèse et historisation des imports

Lors de la validation, une alerte informe l'utilisateur du nombre de données pouvant être intégrées à la synthèse, et du nombre de données non conformes (*gn\_is\_valid=f*) ne pouvant pas l'être. Ces dernières sont alors exportables au format source par l'utilisateur (qui pourra les corriger manuellement et les réimporter) et sont supprimées de la table d'import.

La table `t_source` de la synthèse est ensuite renseignée :

- `name_source` : `gn_import'id_import'`
- `desc_source` : "Données issues du module import - id\_import"
- `entity_source_pk_field` : champ pk table import
-  `validable` : "true" ou "false" selon la configuration du module

Enfin, les données de la table d'import (toutes valides à ce stade) sont poussées vers la synthèse. L'import est historisé (`t_imports`), la procédure est achevée.

## Options

### Option 1 : Réconciliation automatique des noms latins et de leur cd\_nom dans le taxref

L'outil GeoNature, utilise le cd\_nom comme identifiant du taxon. Fréquemment, les données source n'intègrent pas ce cd\_nom mais uniquement un nom latin.

Si cette option était développée, le mapping des champs devrait tenir compte de ce point en permettant de ne pas renseigner le cd\_nom (obligatoire), mais un nom latin source à la place. Le module d'import serait alors chargé d'établir la correspondance entre un nom latin et son cd\_nom dans le taxref, de la manière la plus fiable possible.

Ce rattachement pourrait être développé soit en mettant en place une fonction dans la base de données (ébauche : <https://github.com/PnX-SI/GeoNature/issues/445>), soit en s'appuyant sur l'outil Taxref Fuzzu Match du MNHN (<https://taxref.mnhn.fr/taxref-web/api/doc>).

Ce rattachement automatique reste une source d'erreur importante et s'avère sensible. Quand deux cd\_nom correspondent pour un nom latin donné, alors le module ne devra pas faire de réconciliation, qui restera manuelle et à la charge de l'utilisateur (les données en question seront alors jugées invalides).

### Option 2 : Modification de l'emplacement des tables d'imports

Cette option consiste à renseigner au module la destination de la table d'import de manière dynamique. Dans la base de données, l'administrateur aurait ainsi la possibilité d'organiser les tables d'imports dans les schémas qui lui conviennent, tout en gardant la possibilité de compléter ces tables lors de nouveaux imports. Le fonctionnement imaginé impliquerait une bibliothèque des tables d'import dans la base de données (bib\_import\_table) contenant un identifiant id\_table\_import et son emplacement (table\_import : gn\_import.matable par exemple) renseignable par l'administrateur. Une correspondance sera faite par ailleurs entre un jeu de données, son id\_table\_import, son cd\_mapping, son SRID et les identifiants des différents imports effectués dessus.

### Option 3 : Rédaction d'une documentation utilisateurs

Les documentations destinées aux administrateurs du système sont obligatoires (installation, paramétrage et configuration). Optionnellement, une documentation à destination des utilisateurs peut être prévue dans le cas où le module le nécessite sur les points les moins intuitifs de la manipulation.

## Modalité de mise en oeuvre

### **Ressources disponibles :**

Un certain nombre de documents sont disponibles à l'adresse suivante :

<https://geonature.readthedocs.io/fr/latest/index.html>

En particulier, une documentation à l'attention des développeurs est disponible et doit être consultée, à l'adresse suivante :

<https://geonature.readthedocs.io/fr/latest/development.html>

L'ensemble du code source de l'application GeoNature est disponible sur le GitHub des Parcs nationaux :

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

La documentation d'import de données :

<https://geonature.readthedocs.io/fr/latest/import-level-2.html>

La documentation décrivant les fonctions implémentées dans GeoNature :

<https://geonature.readthedocs.io/fr/latest/admin-manual.html#fonctions>

Des xemples d'imports de données et d'utilisation des fonctions :

<https://github.com/PnX-SI/Ressources-techniques/tree/master/GeoNature/migration>

La documentation relative aux API du MNHN sur lesquelles le module peut s'appuyer :

<https://taxref.mnhn.fr/taxref-web/api/doc>

Le code de l'application TaxHub (taxonomie) est disponible à l'adresse suivante :

<https://github.com/PnX-SI/TaxHub>

Le code de l'API nomenclatures de GeoNature est disponible à cette adresse :

<https://github.com/PnX-SI/Nomenclature-api-module>

Le code de UsersHub (utilisateurs) est disponible à cette adresse :

<https://github.com/PnEcrins/UsersHub>

### **Modalités de développement :**

L'ensemble des développements devra se faire sur la forge GitHub des Parcs nationaux (PnX-Si) dans le dépôt gn\_module\_import.

Les langages utilisés et interfaces devront être en cohérence avec le reste de la suite applicative (Python/Flask/Angular 4/PostgreSQL).

Durant la période de développement, un serveur test devra être accessible et permettra au maitre d'ouvrage et certains partenaires identifiés de suivre les développements et tester les fonctionnalités en place (Flavia APE, parcs nationaux et CBN notamment, et autres utilisateurs en attente du module). Il devra respecter les pré-requis existants pour l'application GeoNature V2.

## Livrables et modalités de livraison

### Code :

L'ensemble du code développé devra être livré sur le dépôt GitHub du module. Le code sera publié sous licence libre comme le reste des applications de la suite GeoNature.

L'ensemble des paramètres liés à l'installation devront être renseignés dans un unique fichier de "settings" du module, et l'ensemble des règles portant sur les fonctionnalités paramétrables du module devront être définies dans un fichier de configuration.

Une cohérence dans le développement devra être observée avec le reste de l'application.



Le fichier de configuration, pouvant être mis à jour sans réinstallation du module,

- répertoire de stockage des fichiers uploadés
- taille maximale des fichiers autorisés pour l'upload
- les données issues du module sont validables (true/false)

### Documentation :

Outre le code, une documentation devra impérativement être produite et rendue disponible pour tous les utilisateurs. La documentation devra à minima détailler les procédures d'installation du module, l'ensemble de sa configuration et des paramètres configurables, et toute autre information utile à l'administration du module à destination de l'administrateur de l'instance.

Une documentation à destination des utilisateurs, détaillant les fonctionnalités du module et les étapes à suivre pour l'import de données peut être prévue de manière optionnelle.

La documentation devra être livrée au format .rst - avec ou sans images/gif - dans un répertoire "docs", sur le dépôt GitHub du module.

## Calendrier

Lancement de la prestation : souhaitée dès que possible  
Délais de livraison à discuter avec le prestataire, horizon aout 2019.

### Modalités de paiement :

- Un acompte de 30% sera versé après validation du devis
- Un acompte de 30% sera versé au cours de la mission, selon le degré d'avancement des développements (fonctionnalités principales mises en place)
- Un solde de 40% sera versé après livraison du module

Les délais de paiements, de quelques semaines, seront à discuter avec le prestataire sélectionné.

## Contacts

Le maître d'ouvrage est l'association Flavia APE. Le contact privilégié pour tout échange lié à la commande est :

**M. Donovan Maillard**  
**[pole.invertebres@gmail.com](mailto:pole.invertebres@gmail.com)**  
**06 61 33 64 56**

A défaut, le référent suppléant pour la prestation est :

M. Yann Baillet  
[asso.flavia@gmail.com](mailto:asso.flavia@gmail.com)

Les développeurs des Parcs nationaux, à l'initiative de l'ensemble applicatif, seront également associés à la mise en œuvre de cette mission afin de s'assurer de la compatibilité du module avec le reste des outils, et s'assurer que le module est développé en cohérence avec le cœur de GeoNature (langages, logique etc).

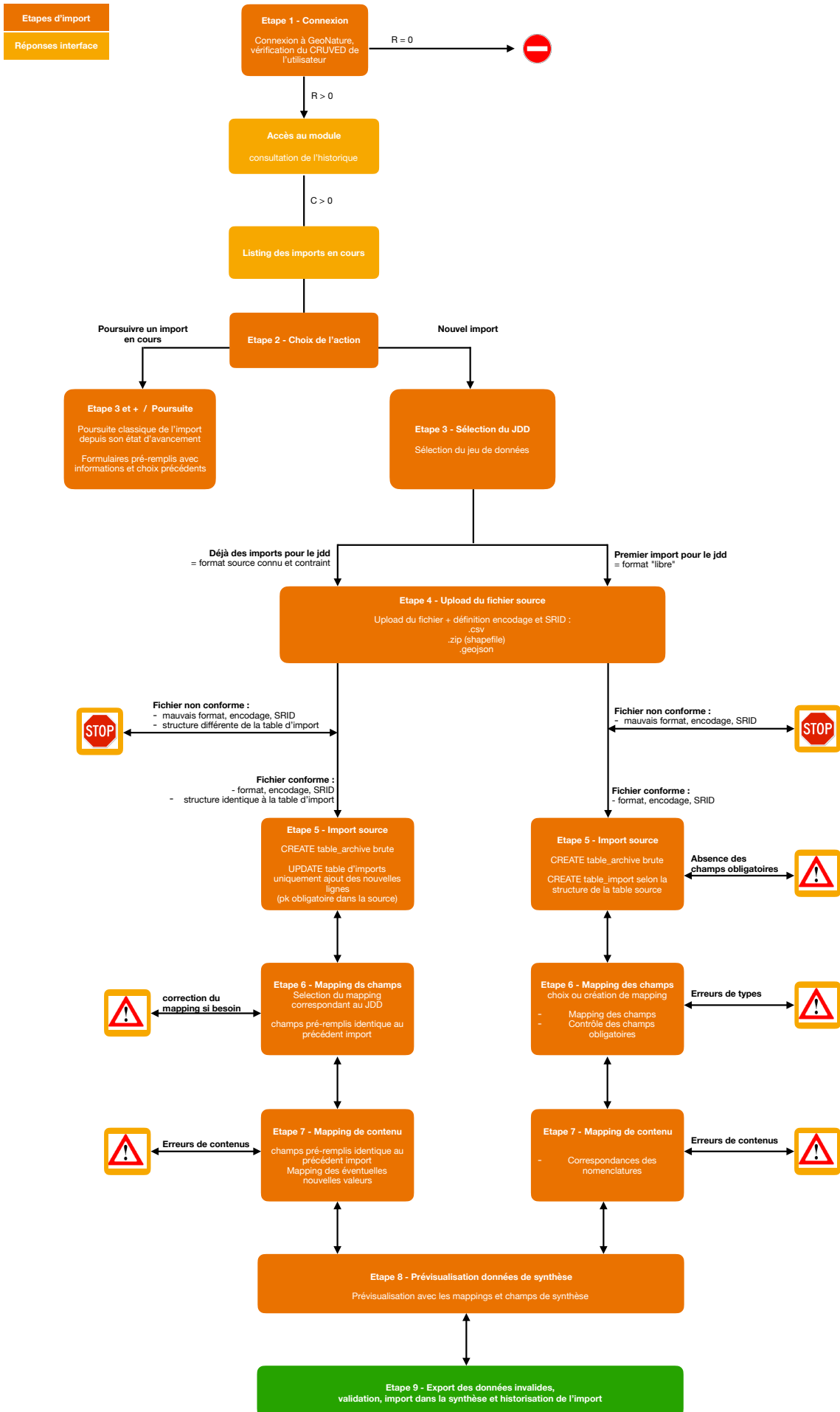
Ils peuvent être sollicités par l'intermédiaire de Camille Monchicourt (Parc national des Ecrins) :

[camille.monchicourt@ecrins-parcnational.fr](mailto:camille.monchicourt@ecrins-parcnational.fr)  
04 92 40 20 30

## Modalités de réponse

Si la mission vous intéresse, merci de retourner vos propositions techniques et financières par mail à l'adresse **[pole.invertebres@gmail.com](mailto:pole.invertebres@gmail.com)** avant le mercredi 6 mars 2019 à 17h.

# Annexe 1 : Schéma du mécanisme et étapes de l'import



## Annexe 2 : Ebauche de MCD pour le module

Cette ébauche ne vise pas à définir un MCD définitif, mais à illustrer le fonctionnement imaginé pour le module, et ses interactions avec le reste de la base de données de GeoNature. Il devra être abondé et amélioré au cours de la prestation.

