



BiomMap:
développements &
adaptations d'une
application
GeoNature-citizen

Documentation technique

Champs-Libres

Version 31 octobre 2021

Table des matières

1	Interface d'administration	3
1.1	Gestion des utilisateurs	3
1.2	Gestion des programmes	3
1.2.1	1) Projets	3
1.2.2	2) Zones géographiques	4
1.2.3	3a) Formulaires dynamiques	4
1.2.4	3b) Types de sites	4
1.2.5	4) Programmes	5
2	Configuration de l'application et édition de contenu	5
2.1	Page du tableau de bord	5
2.1.1	Plan de la page	5
2.1.2	Données du tableau de bord	6
2.1.3	Sources des données	6
2.2	Editer le contenu statique	6
2.3	Modifier l'image d'un programme	7
2.4	Configuration du serveur SMTP	7
3	Import / requalification de données externes	8
3.1	Introduction	8
3.2	Fonctionnement du script d'import	8
3.3	Correspondance entre propriétés	8
3.4	Import des arbres & haies remarquables de Wallonie	9
4	Utilisation des données encodées	10
4.1	Utilisation via l'API	10
4.1.1	Détails de points d'API les plus utiles	10
4.1.2	Utilisation de ces points d'API avec QGIS	11
4.2	Accès à la base de données	12
4.3	Interaction entre QGIS et la base de données	12
4.3.1	Principe	12
4.3.2	Accès	13

Table des figures

1	utilisation d'un point d'API dans QGIS	11
2	Projet QGIS "lecture biommap"	11
3	Screenshot de pgweb	12
4	Connexion à une base de données PostGIS avec QGIS	13
5	Liste des couches PostGIS	13

Liste des tableaux

1 Interface d'administration

L'interface d'administration permet principalement de gérer les programmes, mais sans les données encodées. Elle est disponible sous l'url `api/admin`, soit <https://biomap.champs-libres.be/api/admin/>.

1.1 Gestion des utilisateurs

Dans l'interface d'administration, le deuxième onglet, "Utilisateurs", permet de visualiser et de gérer les utilisateurs inscrits sur l'application Biommap.

Cette page liste tous les utilisateurs inscrits. Pour chaque utilisateur, lorsqu'on clique sur l'icône d'édition, on peut changer les caractéristiques de l'utilisateur : nom, prénom, nom d'utilisateur, email, téléphone, organisation, une image d'avatar, et 2 cases à cocher pour 1) activer ou non l'utilisateur et 2) donner les droits d'administrateur.

La création d'un utilisateur se fait via l'application Biommap via échanges d'email (après avoir cliqué sur "Créez votre compte"). Un serveur SMTP doit être branché dans l'application pour cela. A priori, l'interface d'administration des utilisateurs n'est pas utilisée pour ajouter des utilisateurs.

1.2 Gestion des programmes

Dans l'interface d'administration, le troisième onglet, "Enquêtes", permet d'ajouter et de gérer les programmes d'observations de l'application. Cette accès ne permet pas de gérer les données encodées par les utilisateurs ou importées : pour cela, il faut utiliser la connexion avec QGIS ou directement via la base de données.

L'interface d'administration est accessible via l'adresse `/api/admin`. L'onglet "Enquêtes" se subdivise en 5 parties qui permettent d'ajouter ou d'éditer un programme :

- 1) Projets
- 2) Zones géographiques
- 3a) Formulaires dynamiques
- 3b) Types de sites
- 4) Programmes

Ces onglets doivent être remplis dans cet ordre pour faire un programme car les programmes ont besoin des autres entités (projets, zones géographiques, formulaires, types de sites).

1.2.1 1) Projets

Création d'un projet. Un projet peut contenir plusieurs programmes. Il faut au moins un projet pour ajouter des programmes. Tous les programmes peuvent dépendre du même projet : lors du prototype, un seul projet a été créé.

1.2.2 2) Zones géographiques

Définition d'une zone géographique pour les programmes. Une zone géographique doit avoir un nom et un fichier geojson ou kml de type Polygon ou Multi-Polygon dans le système de coordonnées WGS84 (EPSG :4326). Typiquement, ce fichier géographique est un polygone représentant le territoire d'une commune.

Pour obtenir facilement un fichier d'une zone géographique, on pourra utiliser un découpage administratif (e.g., [découpage des communes en Belgique](#)) ou bien à partir d'OpenStreetMap via une requête overpass.

Par exemple, voici la requête utilisée pour retrouver la commune de Wasseiges via overpass-turbo.eu : <http://overpass-turbo.eu/s/15qX>. Lorsque la requête est lancée, on peut télécharger le polygone des limites communales en cliquant sur "Exporter".

```
[out:json][timeout:25];
{{geocodeArea:Wasseiges}}->.searchArea;
(
  relation["admin_level"="8"](area.searchArea);
);
out body;
>;
out skel qt;
```

1.2.3 3a) Formulaires dynamiques

Définition des formulaires utilisés dans les rapports de visites. Ces formulaires sont utilisés dans les programmes pour qualifier les observations à l'aide d'un rapport de visite.

Ils comprennent un nom et le formulaire en tant que tel sous forme d'un schéma json.

Au moment d'écrire ces lignes, il n'y a pas de documentation exhaustive pour construire ces formulaires, mais les exemples fournis permettent de créer de nouveaux formulaires. En bref :

- les "steps" permettent de décomposer le formulaire en différentes étapes dans le rapport de visites, correspondant à différents écrans.
- les "properties" définissent les données qui peuvent être encodées. Ces "properties" ont différents "types" : string, integer, ...
- pour chaque "properties", le type de sélecteur peut être défini : "type" : "select" avec un tableau de choix ("titleMap"), "type" : "textarea", etc.

NB : par défaut, les photos sont intégrées dans chaque rapport de visite, sans qu'il faille les ajouter au formulaire.

1.2.4 3b) Types de sites

Définition des types de sites. Chaque programme doit être qualifié par un ou plusieurs type de site. Par exemple, le programme "Bosquets, buissons, friches & lisières" est qualifié par les sites "Lisières", "Friches", "Buissons", "Bosquets".

Lors de l'ajout d'un site dans un programme, il est demandé de définir le type de site (avant l'encodage d'un rapport de visite).

Chaque type de site comporte un nom (sous "Type"), une catégorie ("Category"), un formulaire ("Custom Form") et en option un pictogramme.

1.2.5 4) Programmes

Définition des programmes d'observations. C'est l'entité finale qu'il faut définir pour voir apparaître un programme dans l'application. Nous détaillons ici un programme de type "site" :

- module : "sites" ou "observations"
- custom form : formulaire dynamique défini en 3a) pour les rapports de visites
- geometry : zone géographique défini en 2). Le programme ne pourra être utilisé que dans cette zone.
- project : projet défini en 1).
- title : titre du programme, apparaissant dans l'application
- short desc : description courte, apparaissant dans l'application
- long desc : description longue. Celle-ci n'est pas utilisée dans l'application.
- geometry type : type de géométrie que l'on veut éditer. Cette géométrie peut être de 3 types : POINT, LINESTRING (ligne) ou POLYGON. Ce champ détermine le type d'encodage des géométries du programme. Il n'est pas possible de combiner plusieurs types de géométries dans un programme.
- image ou logo : chemin relatif ou absolu vers une image pour définir l'image du programme apparaissant en médaillon (voir Ajouter/changer une image d'un programme). Exemple : ../api/media/lisiere.jpg
- types de site : ajouter un ou plusieurs types de sites

2 Configuration de l'application et édition de contenu

2.1 Page du tableau de bord

Une nouvelle page "tableau de bord" a été ajoutée à <https://biomap.champs-libres.be/fr/dashboard>.

2.1.1 Plan de la page

- 1) Données de base
- 2) Occupation du sol
- 3) Les linéaires
- 4) Les éléments isolés
- 5) Les éléments surfaciques
- 6) Le suivi via BiomMap (des données externes)

2.1.2 Données du tableau de bord

Données statiques Une partie des données du tableau de bord sont statiques : elles n'ont pas pu être trouvées via des services web. Ces données statiques sont stockées dans un objet JS dans le fichier `dashboardData.ts`. Par exemple, il y a la superficie et le nombre d'habitants de la commune.

Données issues des programmes

2.1.3 Sources des données

1) Données de base Les données de base telles que la superficie de la commune et le nombre d'habitants sont disponibles pour chaque commune sur la page web de STATBEL via <https://statbel.fgov.be/fr/commune/wasseiges#dashboard1> (exemple pour Wasseiges). Les chiffres pour toutes les communes de Belgique sont accessibles dans [ce fichier](#) accessible dans [cette page](#).

2) Occupation du sol Des données d'occupation du sol agrégées par commune sont disponibles via STATBEL sur [cette page](#), en particulier [ce fichier](#). Ce sont ces données qui ont été utilisées.

Par ailleurs, il y a aussi des données d'occupation du sol en Wallonie :

- le Plan de Secteur : accessible [ici](#). Toutefois, cette donnée renseigne la situation juridique des terrains, mais n'indique pas l'occupation actuelle des sols. Par exemple, un terrain constructible peut être occupé par une prairie.
- WALOUS : accessible [ici](#) mais uniquement en téléchargement. Cette donnée est récente (2018) et le plus proche de ce qu'on entend par occupation du sol. Elle n'est malheureusement pas disponible en services web.
- COSW 2007 : accessible [ici](#). Cette donnée a été dépréciée par le produit WALOUS.

3) Production agricole et rendements Les données de production, surface emblavées et rendements sont accessible chaque année via [cette page](#). Les données par commune pour 2020 ont été téléchargées [ici](#).

2.2 Editer le contenu statique

Par contenu statique, nous entendons toutes les pages dont le contenu n'est pas modifié par les utilisateurs du site, soit les éléments de la page d'accueil (images, footer) et la page "À propos". Il s'agit aussi de la plupart des images et logos utilisés dans l'application.

Il n'y a pas encore d'interface d'édition simple pour ce contenu statique à la manière d'un CMS. La marche à suivre est la suivante :

- modifier ces éléments dans le dossier `patches/frontend`

- recompiler l'application front-end (`npm run build:i18n-ssr`)

Les images dans les sous-dossiers `patches/frontend/src/assets` peuvent être simplement remplacées par de nouvelles. Pour le contenu HTML du footer et de la page "À propos", il faut éditer les fichiers `patches/frontend/src/custom/footer/footer.html` et `patches/frontend/src/custom/about/about.html`. Il existe également un fichier `patches/frontend/src/custom/home/home.html` qui permet d'ajouter une zone dans la page d'accueil.

2.3 Modifier l'image d'un programme

Pour changer l'image médaillon d'un programme, qui apparait lorsqu'on clique sur l'entrée du menu "Programmes", il faut :

1) Servir l'image

- Placer l'image au format jpg ou png dans le dossier `patches/frontend/src/assets/mon_image.jpg`
- Recompiler le conteneur `media:docker-compose build --no-cache media`
- l'image est servie sur `/api/media/mon_image.jpg`

2) Indiquer l'url de l'image dans les settings du programme

Dans l'interface d'administration, aller sur Enquêtes > Programmes. Editer le programme voulu. Dans le champ Image, ajouter simplement : `../api/media/mon_image.jpg`, sans guillemets.

2.4 Configuration du serveur SMTP

L'envoi d'email est nécessaire pour que les utilisateurs puissent s'enregistrer. Il se fait simplement en configurant les paramètres suivants dans le fichier `patches/back/default_config.toml`.

```
MAIL_USE_SSL = false
MAIL_STARTTLS = true
MAIL_HOST = 'mail.myemaildomain.org'
MAIL_PORT = 587
MAIL_AUTH_LOGIN = 'myself@myemaildomain.org'
MAIL_AUTH_PASSWD = '*****'
```

ainsi que :

```
URL_APPLICATION = https://biomap.champs-libres.be/
```

Le dernier paramètre est utilisé pour construire le lien de confirmation qui est envoyé à l'utilisateur afin de valider son inscription.

3 Import / requalification de données externes

3.1 Introduction

Pour certains programmes, il est possible d'importer des données externes à GeoNature et de pouvoir les requalifier dans l'application GeoNature-citizen. L'intérêt est que cela permet de profiter d'un encodage déjà existant et qu'une partie des propriétés peuvent être directement importés, sans qu'il soit nécessaire de les mesurer/observer. Par exemple, nous avons importé des données issues du géoportail wallon indiquant les [arbres remarquables](#) dans le programme "Arbres" de Biommap. Des propriétés telles que la hauteur de l'arbre et la circonférence ont pu être importées. Nous avons également investigué l'import de données OpenStreetMap (arbres et haies).

A terme, cela permettrait d'améliorer des jeux de données existants, d'une part en ajoutant de nouveaux éléments et d'autre part en mettant à jour ou ajoutant des propriétés manquantes d'un élément existant. Biommap permet aussi de qualifier de manière plus riche ces données, notamment en permettant d'y ajouter des photographies.

3.2 Fonctionnement du script d'import

La documentation technique pour utiliser le script d'import est écrite dans le dépôt (dossier `utils`). Le script d'import est un script Python qui doit tourner dans un environnement virtuel. La commande est :

```
(venv) python3 geojson2sql.py
```

Ce script doit être adapté pour chaque source de données externes! Le script prend en entrée un fichier géographique de données externes au format geoJSON en WGS84 (EPSG :4326).

Le script génère des instructions SQL pour alimenter **deux** tables différentes : la table des sites et la table des visites.

3.3 Correspondance entre propriétés

Pour chaque propriété intéressante dans la donnée externe que l'on veut importer dans un programme GeoNature-citizen, il faut obligatoirement 1) que cette propriété existe dans le formulaire d'encodage et 2) réaliser une correspondance (mapping) entre le nom de cette propriété dans la couche externe et le nom de la propriété dans le formulaire d'encodage.

Le script d'importation crée des entrées dans la base de données en ne remplissant pas tous les champs de la table. Seuls les champs obligatoires (non-nullables) et les champs informatifs sont remplis.

Certains champs doivent être obligatoirement remplis, comme :

- les champs `id_program` et `id_type` DOIVENT correspondre aux bonnes valeurs! Le premier champ décidera à quel programme les sites seront rattachés.

- le champ `id_site` dans la table `sites` et `visit`
- le champ `timestamp_create` dans la table `sites` et `visit`
- le champ `geom` dans la table `sites` et contient la géométrie.
- le champ `uuid_sinp` dans la table `sites` qui doit être un identifiant unique `uuid`

D'autres champs non obligatoires mais utiles sont :

- le champ `name` qui donnera le nom de l'élément.
- le champ `obs_txt` qui donnera le nom de l'utilisateur.

Certaines valeurs dans le script seront converties automatiquement :

- la valeur **"ID"** sera convertie en un identifiant unique pour le site qui sera créé en ajoutant 1000000 à l'index de la table `geojson`.
- La valeur **"UUID"** sera convertie en une formule qui crée une valeur `uuid`.
- la valeur **"POINT"** ou **"LINESTRING"** sera convertie en une formule PostGIS `ST_GeomFromText` basée sur les coordonnées du fichier `geojson`
- la valeur **"NOW"** sera convertie en une formule qui crée un horodatage (pour les champs `timestamp`).

Enfin, si les valeurs commencent par `properties.`, elles seront mises en correspondance avec une propriété dans le `geojson`. Par exemple, `properties.COMMENTAIR`.

Pour remplir la table des visites, la même fonction est appliquée. Mais il y a une autre convention spéciale qui s'applique pour construire les détails json de la visite : la valeur **"JSON"** sera convertie en une formule dédiée qui construira l'objet json dans la base de données.

3.4 Import des arbres & haies remarquables de Wallonie

La donnée a été téléchargée via la page <http://geoportail.wallonie.be/catalogue/d594f5a3-34ac-4cc2-a357-aae5d5263f35.html> et convertie au format `geoJSON` en WGS84 via QGIS.

Pour mémoire, voici la liste des colonnes et la liste des valeurs correspondantes pour la table des sites :

```
column_list = ['id_site', 'uuid_sinp', 'id_program', 'name', 'geom', 'timestamp_create', 'id_t
values_list = ['__ID__', '__UUID__', '2', 'properties.SPFR', '__POINT__', '__NOW__', '1', 'géc
```

Voici la liste des colonnes et la liste des valeurs correspondantes pour la table des visites :

```
column_list = ['id_site', 'json_data', 'timestamp_create', 'obs_txt']
values_list = ['__ID__', '__JSON__', '__NOW__', 'import']
```

4 Utilisation des données encodées

4.1 Utilisation via l'API

L'application GeoNature-citizen expose les données des programmes, des observations et des sites au travers de points d'API pour le fonctionnement de l'application.

Ces points d'accès peuvent aussi être utilisés pour avoir, sans authentification requise, un accès direct aux données.

4.1.1 Détails de points d API les plus utiles

1) Point d'API de tous les programmes

- url : `api/programs`
- exemple : <https://biomap.champs-libres.be/api/programs>

Expose tous les programmes définis, mais sans les données (observations ou sites) de ces programmes. Peut-être utile pour retrouver l'id d'un programme afin de voir le détail du programme (point 2) ou ses observations ou sites (point 3).

2) Point d'API d'un programme

- url : `api/programs/{id}`
- exemple : <https://biomap.champs-libres.be/api/programs/1>

Expose le détail d'un programme, comme dans le point d'API précédent, sans les données (observations ou sites) de ce programme, mais avec la géométrie de la zone géographique du programme (par exemple la commune).

3) Point d'API des sites du programme

- url : `api/sites/programs/{id}`
- exemple : <https://biomap.champs-libres.be/api/sites/programs/1>

Expose tous les sites encodés d'un programme. Donne un geojson valide qui peut être utilisé dans QGIS.

4) Point d'API du détail d'un site

- url : `api/sites/{id}`
- exemple : <https://biomap.champs-libres.be/api/sites/1>

Expose le détail d'un site. Comprend les photos (noms des fichiers uniquement) et tous les rapports de visites qui contiennent les informations rentrées par les utilisateurs dans le formulaire des visites.

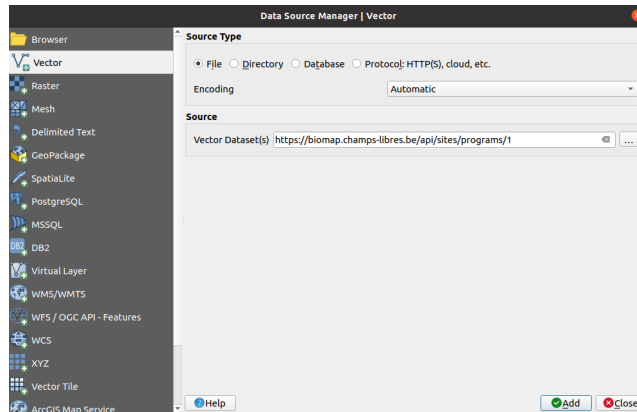


Figure 1 – utilisation d'un point d'API dans QGIS

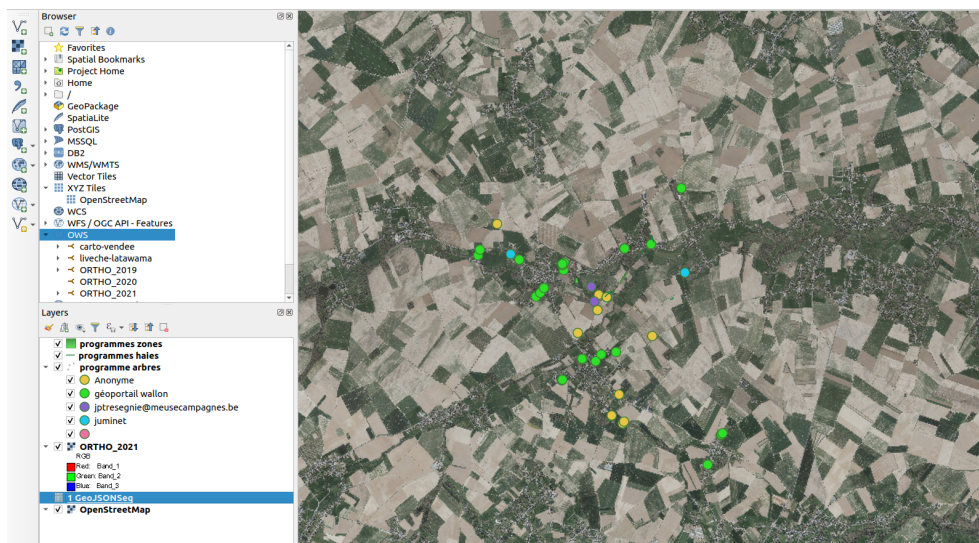


Figure 2 – Projet QGIS "lecture biommap"

4.1.2 Utilisation de ces points d'API avec QGIS

Les données des sites peuvent être directement intégrées dans un projet QGIS. Ajouter une couche vectorielle au projet en tant que fichier en spécifiant l'adresse du point d'API des sites d'un programme, par exemple <https://biomap.champs-libres.be/api/sites/programs/1>. La figure suivante montre une capture d'écran de cet ajout.

Ces points d'API de données par programme sont en lecture seule. Un projet QGIS peut donc être partagé avec ces points d'API sans risquer que l'utilisateur modifie les données. Pour utiliser QGIS en ayant la possibilité d'éditer les données, voir le point "Interaction QGIS - base de données".

Un projet QGIS avec ces points d'API a été préparé. Une capture d'écran de ce projet est dans la figure suivante :

id site	uid	id program	name	id type	geom	timestamp create	timestamp update	id role	obs st	email
1000000	145455a-378b-40b-86a5-6156544b3ca	2	import artres remarquables	1	0101000000E08A705F1A1440F6D216123504940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000001	3a3a4803-502b-4c38-96a5-4b6317e31a0	2	import artres remarquables	1	010100000074E484441A1440F6D216123504940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000002	84659a6-7767-4692-ae15-657982105380	2	import artres remarquables	1	0101000000CD461FF301214405689C4D47504940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
5	48D679-3d78-4a4d-987-403a7ca5590	2	ecozec	1	0101000000769A88B9701640A9C6F985DCE4840	2021-08-25T09:36:46.540384Z	2021-08-25T09:36:46.540384Z	1	jammet	juken.mme@champs-l
6	c04640ac-4443-4484-86c4-546061a8701	2	point	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-25T09:51:30.341872Z	2021-08-25T09:51:30.341884Z	1	jammet	juken.mme@champs-l
7	199466f-6467-49c2-8a18-c0117172871	2	test	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-25T12:17:31.716232Z	2021-08-25T12:17:31.716392Z	1	jammet	juken.mme@champs-l
8	10f4959-4312-451e-8b49-495c3b0d8e8	2	encl246	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-25T15:05:35.021332Z	2021-08-25T15:05:35.021445Z	1	jammet	juken.mme@champs-l
9	4596131-8443-4c44-940b-87909c9807	2	test	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-26T08:52:30.621367Z	2021-08-26T08:52:30.621709Z	1	jammet	juken.mme@champs-l
10	42b13d33-a27c-420c-8b4b-0771e0c9009	2	test	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-26T10:13:58.208071Z	2021-08-26T10:13:58.20838Z	1	jammet	juken.mme@champs-l
1000004	28c27539-1d2c-443c-8410-d1ee0a22716	2	import artres remarquables	1	0101000000159F6F3C0A1404F4E8EBC04F4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
12	14b4064-043c-41d8-899b-97b0c81e194	2	test	1	01010000002E610000E0C0C800891640D888F8E0D4...	2021-08-26T12:11:48.209542Z	2021-08-26T12:11:48.209592Z	1	jammet	juken.mme@champs-l
1000005	88c0566-95ca-4188-899b-97b0c81e194	2	import artres remarquables	1	01010000009F8D1180308144054311880D4E4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000006	a95ca046-848e-4060-918a-c5ccac018d8	2	import artres remarquables	1	01010000006AC817A06A1414054311880D4E4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000007	895a466-4b4f-4b11-474c-61a6a02229f	2	import artres remarquables	1	010100000074E8303281814408D24C388184E4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000008	16871607-0949-4960-8cab-74002f98a2	2	import artres remarquables	1	010100000003C3B8C407F3F1440CF68C1205484940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000009	518a4c3-09a0-4955-6a3c-25a11241313	2	import artres remarquables	1	010100000027E481C321440C9E7F8F864C4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
1000010	6a3a37a-1429-48b0-8a80-35889ba0bd0	2	import artres remarquables	1	010100000074E8303281814408D24C388184E4940	2021-09-15T08:53:19.533946Z	NULL	NULL	import	NULL
19	a16c290-4a07-4a10-402a-67b3a670bdc	1	test	1	01010000002E6100000300000C481919C4C7B16493C471...	2021-08-26T14:15:06.559712Z	2021-08-26T14:15:06.559786Z	1	jammet	juken.mme@champs-l
20	a8dbae1-4949-4a9d-48d8-ba0f01897a	1	test	1	01010000002E610000070000005D4FC025A801640A89AAA...	2021-08-26T14:16:45.268996Z	2021-08-26T14:16:45.268996Z	1	jammet	juken.mme@champs-l
21	09a5023e-a188-4535-4069-2ca0c6487126	1	test BOBO	1	01010000002E6100000000000000000000000000000000...	2021-08-26T16:01:34.702757Z	2021-08-26T16:01:34.702766Z	1	jammet	juken.mme@champs-l
22	8c22454-950-4963-aa80-445a4ca1849	1	ligne	1	01010000002E6100000000000000000000000000000000...	2021-08-27T07:15:41.799182Z	2021-08-27T07:15:41.799213Z	1	jammet	juken.mme@champs-l

Figure 3 – Screenshot of pgweb

4.2 Accès à la base de données

L'accès à la base de données est souhaitable pour gérer directement les données encodées dans l'application. L'interface d'administration (voir Gestion-des-programmes.md) ne permet que la définition d'un programme mais ne permet pas de gérer (créer, modifier, supprimer) les données encodées. Les données peuvent être aussi gérées via un SIG (voir la section "Interaction QGIS-BD").

L'architecture de l'application est susceptible de varier selon les installations, mais si l'application a été installée suivant la configuration par conteneur docker, la base de données est accessible via :

- host : db
- dbname : referentielsdb
- SSL mode : disable

Pour le nom d'utilisateur, le port et le mot de passe, se référer à l'administrateur de l'application.

En outre, il y a un conteneur pgweb qui permet de gérer facilement la base de donnée. Ce conteneur lance une application pgweb qui permet de voir et éditer les tables, effectuer des requêtes, etc. (cfr figure suivante).

Sur l'hébergement de Champs-Libres, l'application pgweb est accessible via : <http://liveche.champs-libres.be:8081/>

4.3 Interaction entre QGIS et la base de données

4.3.1 Principe

Il est possible d'utiliser QGIS pour accéder directement aux données de la base de données BiomMap. Une table d'une base de données peut être ajoutée à un projet QGIS en ajoutant une couche PostGIS, comme indiqué dans la figure suivante.

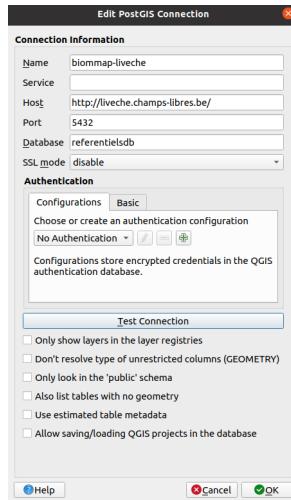


Figure 4 – Connexion à une base de données PostGIS avec QGIS

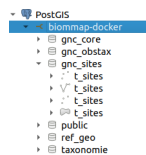


Figure 5 – Liste des couches PostGIS

Il faut rentrer les informations d'authentification (nom d'utilisateur + mot de passe) pour se connecter à la base de données. Attention, cela ne peut fonctionner que si la base de données de GeoNature-citizen/Biommap a été exposée à l'extérieur, contacter l'administrateur pour le savoir.

Ensuite, toutes les tables sont affichées dans le navigateur (browser) de données de QGIS, comme indiqué dans la figure suivante. Les tables sont organisées selon plusieurs schémas (càd des groupes de tables) : gnc_core, gnc_sites, ... Les tables des sites avec leur information sont dans le schéma "gnc_sites". Comme les sites peuvent être de 3 types de géométrie, QGIS sépare la table des sites (t_sites) en plusieurs couches. Les tables sans géométries ne sont pas listées ici, mais elles peuvent l'être si on a coché "Also list table with no geometry" dans la fenêtre de connexion PostGIS (voir la première figure).

4.3.2 Accès

Il existe un projet QGIS qui permet de lire les données de la base de données et qui a dû être partagé avec la documentation. Ce projet permet non seulement de visualiser et traiter les informations, mais aussi de les éditer (modification, suppression). Attention aux manipulations car QGIS agira dans ce projet directement sur la base de données de Biommap!

Attention, ce projet ne peut fonctionner que si l'accès à la base de donnée a été ouvert à l'extérieur et si les informations de connection (nom d'utilisateur + mot de passe) sont connus.