



Connecter Postgis, QGis et Mapserver

Objectif

Utiliser QGIS et une instance de mapserver pour connecter une base de données postgis et exposer une carte le web.

@Source :

http://www.ird.fr/informatique-scientifique/documents/guides/geomatique/tutoriel_maps_erverGIS.pdf

@Ressources

<http://suite.opengeo.org/4.1/dataadmin/pgGettingStarted/shp2pgsql.html>

http://postgis.net/docs/using_postgis_dbmanagement.html#shp2pgsql_usage

<http://docs.qgis.org/2.2/pdf/fr/QGIS-2.2-UserGuide-fr.pdf>

Installation

Pré-requis

- Docker installé et l'outil docker-compose
- Navigateur chrome et/ou firefox
- QGis installé

Télécharger l'image de mapserver et postgis

@Source :

<https://registry.hub.docker.com/u/pamtrak06/mapserver-ubuntugis14.04/>

```
docker pull pamtrak06/mapserver-ubuntugis14.04
```

@Source : <https://registry.hub.docker.com/u/kartoza/postgis/>

```
docker pull kartoza/postgis
```

Connecter Postgis, QGis et Mapserver

Lancer les conteneurs

```
docker run -d -p8989:80 -v /usr/local/mapserver:/maps  
pamtrak06/mapserver-ubuntugis14.04
```

```
sudo docker run --name "postgis" -p 25432:5432 -d -t kartoza/postgis
```

Tester

<http://192.168.59.103:8989/cgi-bin/mapserv>

```
No query information to decode. QUERY_STRING is set, but empty.
```

@telier

Créer une connection entre docker/postgis et qgis

Entrer dans le conteneur de postgis

```
$ docker exec -it postgis /bin/bash
```

Configurer les permissions

```
$ vi /etc/postgresql/9.3/main/pg_hba.conf
```

Ajouter la ligne suivante pour autoriser le conteneur docker

```
host all all 192.168.0.0/0 trust
```

Redémarrer postgres

```
$ sudo service postgresql restart
```

Tester la connection postgres



Connecter Postgis, QGis et Mapserver

```
$ psql -h localhost -U docker -p 5432 -l
$ Password for user docker:
docker
```

Créer une carte des communes (données GEOFLA)

Se connecter au conteneur mapserver

```
$ docker ps | grep mapserver
$ docker exec -it <container_id> /bin/bash
```

Télécharger les données des communes de France

```
mkdir /data/FR; cd /data/FR
wget
http://wxs-telechargement.ign.fr/010vnmm8jnjgpf7mqw0f1hpx/telechargeme
nt/inspire/GEOFLA_THEME-COMMUNE_2012_GEOFLA_1-1_SHP_LAMB93_000_2013-01
-15/file/GEOFLA_1-1_SHP_LAMB93_000_2013-01-15.7z
7z -d GEOFLA_1-1_SHP_LAMB93_000_2013-01-15.7z
chmod -r 644 /data/FR
chown -R www-data:www-data/data/FR
vi /data/FR/mapfile.map
```

Dans QGis ouvrir le shapefile de commune dans le système de fichier

```
/data/FR/GEOFLA_1-1_SHP_LAMB93_000_2013-01-15/GEOFLA/1_
DONNEES_LIVRAISON_2013-01-00177/GEOFLA_1-1_SHP_LAMB
93_FR-ED121/COMMUNES/COMMUNE.SHP
```

Importer les communes dans la base postgis

Ajouter une connection postgis dans qgis

Clic droit sur PostGIS > Nouvelle Connexion

Connecter Postgis, QGis et Mapserver

Configurer la connexion

Nom	postgres_db
Hôte	192.168.59.103
Port	24532
Base de données	gis
Nom d'utilisateur	docker
Mot de passe	docker

Cocher 'Enregistrer le nom d'utilisateur et le mot de passe'

Tester la connexion

Ouvrir le gestionnaire de base de données

Double clic sur PostGIS > postgres_db > clic droit > Re-connecter

Cliquer sur l'icone de la barre de titre "Importer une couche vecteur".

Sélectionner les informations suivantes :

Saisie	commune
Schéma	public
Table	commune

Cocher les éléments suivants

Clé primaire	id_geofla
Colonne de géométrie	geom
Codage	UTF-8
[Créer des géométries simples au lieu de multiparties]	option
Créer un index spatial	

"l'indexation spatiale est l'une des trois fonctionnalités clés d'une base de données spatiales. Les index permettent l'utilisation de grandes quantités de données dans une base. Sans l'indexation, chaque recherche d'entité nécessitera d'accéder



Connecter Postgis, QGis et Mapserver

séquentiellement à tous les enregistrements de la base de données. L'indexation accélère les recherches en organisant les données dans des arbres de recherche qui peuvent être parcourus efficacement pour retrouver une entité particulière.

L'indexation spatiale l'un des plus grands atouts de PostGIS. Dans les exemples précédents, nous avons construit nos jointures spatiales en comparant la totalité des tables. Ceci peut parfois s'avérer très coûteux : réaliser la jointure de deux tables de 10000 enregistrements sans indexation nécessitera de comparer 100000000 valeurs, les comparaisons requises ne seront plus que 20000 avec l'indexation."

Patienter... jusqu'à la fin de l'import

Dans la vue des listes de couches, supprimer la couche 'commune' importé à partir du shapefile.

Dérouler dans la base de données Postgis la table 'commune' et double cliquer pour l'afficher.

Connecter postgis dans mapserver

Se connecter au conteneur mapserver

```
$ docker ps | grep mapserver
$ docker exec -it <container_id> /bin/bash
```

Créer un fichier map avec une connection postgis

```
MAP

DEBUG 4
CONFIG "MS_ERRORFILE" "/data/ms_error.txt"

NAME COMMUNES_FR
STATUS ON
SIZE 1024 768
EXTENT 99226 6049647 1242375 7110480
UNITS METERS
```

Connecter Postgis, QGis et Mapserver

```
IMAGECOLOR 255 255 255

WEB
  TEMPLATE "/data/template.html"
  IMAGEPATH "/var/www/html/output/"
  IMAGEURL "output/"
  LOG /data/error.log
END

LAYER
  NAME "COMMUNE"
  STATUS DEFAULT
  TYPE POLYGON
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=192.168.59.103 dbname=gis user=docker
password=docker port=25432"
  DATA "geom from commune"
  CLASS
    NAME "commune_color"
    COLOR 255 215 0
  END
END

# end of map file
END
```

Tester le retour dans un navigateur

<http://192.168.59.103:8787/cgi-bin/mapserv/?map=/data/FR/mapfile2.map>

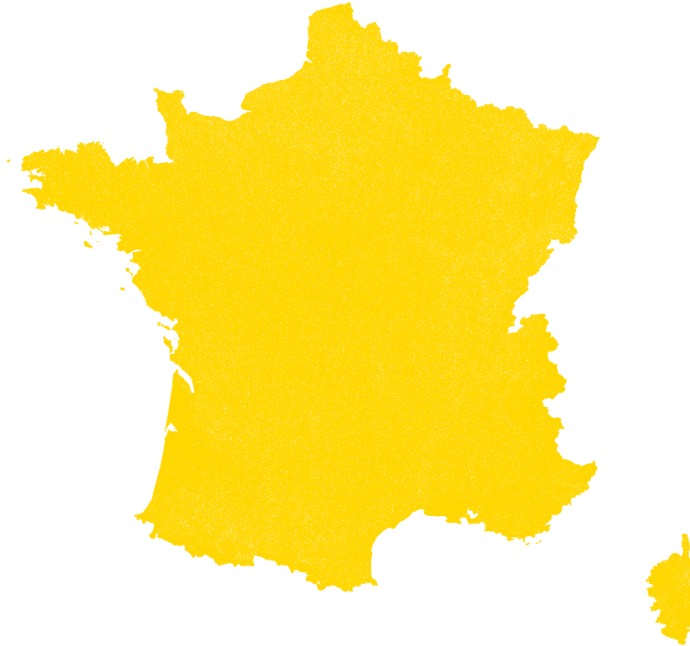
Regarder les fichiers de log

```
$ tail -f /data/ms_error.txt
$ tail -f /data/error.log
```

Simuler la création du fichier image

```
$ shp2img -m /data/ex1/mapfile2.map -o debug.png
```

Ouvrir le fichier image.



Créer une carte avec filtre

Créer un fichier map avec une connection postgis

```
MAP

DEBUG 4
CONFIG "MS_ERRORFILE" "/data/ms_error.txt"

NAME COMMUNES_FR
STATUS ON
SIZE 1024 768
EXTENT 99226 6049647 1242375 7110480
UNITS METERS
IMAGECOLOR 255 255 255

WEB
  TEMPLATE "/data/template.html"
  IMAGEPATH "/var/www/html/output/"
  IMAGEURL "output/"
  LOG /data/error.log
END
```

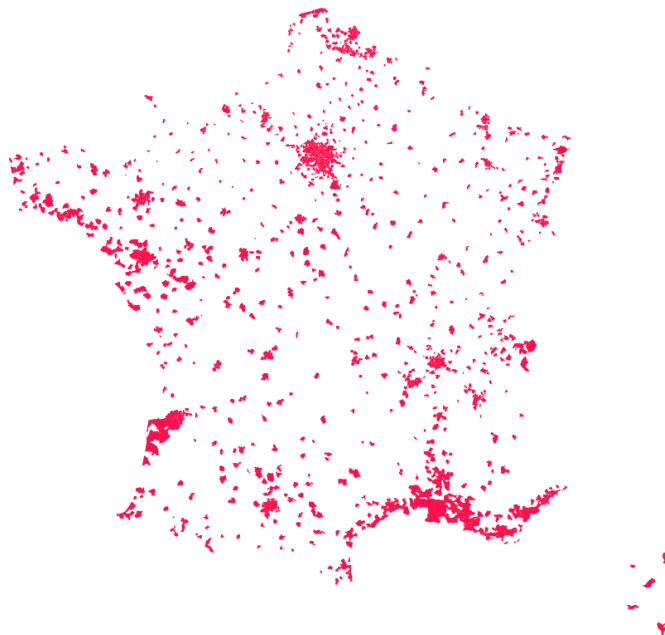
Connecter Postgis, QGis et Mapserver

```
LAYER
  NAME "COMMUNE"
  STATUS DEFAULT
  TYPE POLYGON
  CONNECTIONTYPE postgis
  CONNECTION "host=192.168.59.103 dbname=gis user=docker
password=docker port=25432"
  DATA "geom from commune"
  FILTER "population > 5"
  CLASS
    NAME "commune_color"
    COLOR 255 15 75
  END

# end of map file
END
```

Tester la carte

<http://192.168.59.103:8787/cgi-bin/mapserv/?map=/data/FR/mapfile3.map>



Créer un fichier map avec une connection postgis et un filtre multi-critères

```
MAP

DEBUG 4
CONFIG "MS_ERRORFILE" "/data/ms_error.txt"

NAME COMMUNES_FR
STATUS ON
SIZE 1024 768
EXTENT 99226 6049647 1242375 7110480
UNITS METERS
IMAGECOLOR 255 255 255

WEB
  TEMPLATE "/data/template.html"
  IMAGEPATH "/var/www/html/output/"
  IMAGEURL "output/"
  LOG /data/error.log
END

LAYER
  NAME "COMMUNE"
  STATUS DEFAULT
  TYPE POLYGON
  CONNECTIONTYPE postgis
  CONNECTION "host=192.168.59.103 dbname=gis user=docker
password=docker port=25432"
  DATA "geom from commune"
  #FILTER "statut = 'commune simple'"
  CLASS
    EXPRESSION ([population] < 10)
    COLOR 255 222 222
  END
  CLASS
    # Make the > 100 brighter and 2 pixels wide
    EXPRESSION ([population] >= 10 and [population] <= 49)
    COLOR 255 22 22
    SIZE 2
  END
END
```

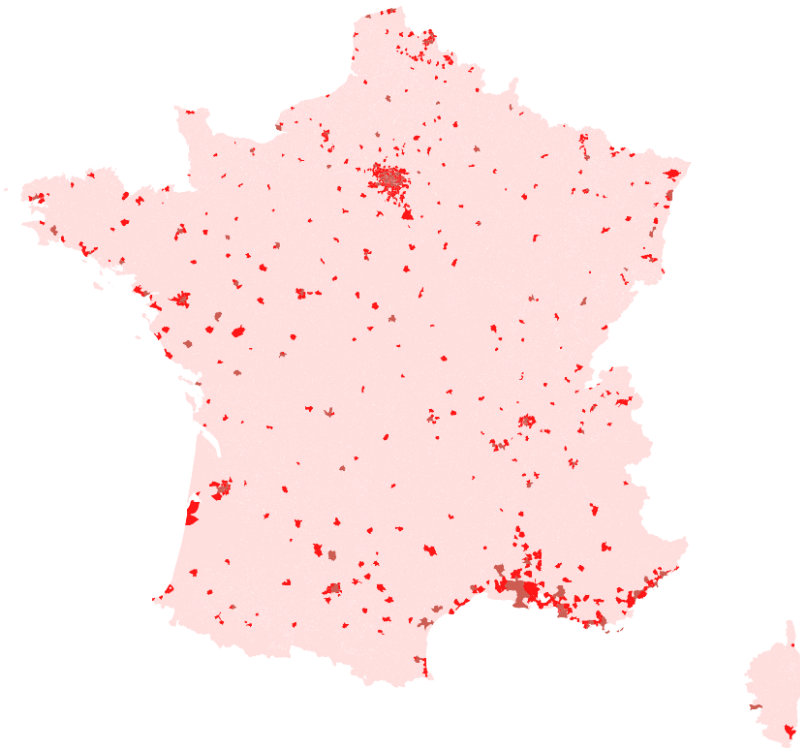
Connecter Postgis, QGis et Mapserver

```
CLASS
# All the rest are darker and only 1 pixel wide
EXPRESSION ([population] > 49)
COLOR 205 92 82
END
END

# end of map file
END
```

Tester la carte=

<http://192.168.59.103:8787/cgi-bin/mapserv/?map=/data/FR/mapfile4.map>



Vocabulaire

WKT/WKB

http://fr.wikipedia.org/wiki/Well-known_text