

R1- l'organisation des bases de données dans Alteris

La base de données postgres du Websig a un miroir d'échanges.

Comme cela GEMINI a la main sur les deux.

`/var/www/alteris-geomatic/data/referentiels`

`home/debian/osm_data/referentiels/`

Organisation de referentiels

Arborescence géographique claire calquée sur celle de l'INSEE et de l'IGN.

Voici l'ordre pyramidal des dossiers

Région (R93 pour la Région PACA)

Département

EPCI (code SIREN)

Commune

Partenaire de code a dit

C'est une excellente décision. Une arborescence structurée est la colonne vertébrale d'un SIG professionnel. En calquant ton stockage sur les codes officiels (INSEE et SIREN), tu garantis l'interopérabilité de tes données avec toutes les futures bases Open Data.

Voici comment nous allons configurer cette structure sur ton VPS pour qu'elle soit à la fois logique pour l'humain et efficace pour le code Node.js.

1. La Structure Physique des Dossiers

Nous allons utiliser les **Codes Insee** (Région, Département, Commune) et le **SIREN** (EPCI) pour nommer les dossiers. C'est plus fiable que les noms alphabétiques qui peuvent avoir des espaces ou des accents.

Exemple pour **Toulon** : Région 93 (PACA) > Département 83 (Var) > EPCI 248300543 (TPM) > Commune 83137 (Toulon)

2. Création de l'arborescence

Tu peux lancer cette commande en SSH pour préparer le terrain pour Toulon :

Bash

```
mkdir -p /var/www/alteris-geomatic/data/referentiels/R93/D83/E248300543/C83137
```

3. Organisation des fichiers GPKG

À l'intérieur de chaque dossier "Commune", nous stockerons les fichiers selon une nomenclature stricte :

- `C83137_contour.gpkg` : La limite administrative (IGN).
- `C83137_filaires.gpkg` : Tes extractions de voies spécifiques si besoin.
- `C83137_iris.gpkg` : Le découpage infra-communal de l'INSEE (très utile pour tes futurs calculs de densité).

4. Pourquoi cette structure est vitale pour ton code ?

En structurant ainsi, nous allons pouvoir automatiser les requêtes. Par exemple, si tu sélectionnes Toulon sur l'interface, le serveur saura instantanément où chercher : `path = "/data/referentiels/R" + reg_code + "/D" + dep_code + "/E" + siren + "/C" + insee + "/"`

5. Intégration dans PostgreSQL

Même si les fichiers sont rangés dans des dossiers, nous allons les importer dans des tables structurées de la même manière dans `alteris_db`. Je te suggère de créer une table maître pour les communes :

SQL

```
CREATE TABLE referentiel_communes (  
  id SERIAL PRIMARY KEY,  
  code_insee VARCHAR(5) UNIQUE,  
  nom_commune TEXT,  
  code_epci VARCHAR(9),  
  code_dept VARCHAR(3),  
  code_reg VARCHAR(2),  
  geom geometry(MultiPolygon, 4326)  
);  
  
CREATE INDEX idx_communes_geom ON referentiel_communes USING GIST(geom);
```

Prochaine action pour toi :

1. **Transférer ton GPKG** de Toulon dans le dossier `/C83137/` que nous venons de créer.
2. **Me donner le nom exact du fichier** pour que je te génère la commande d'importation `ogr2ogr` qui remplira automatiquement les colonnes de code (Insee, EPCI, etc.).

Une fois cela fait, nous pourrons passer à l'étape supérieure : **croiser ce contour de Toulon avec ta matrice de "Profondeur du motif urbain"** (les buffers de 30m, 80m, 150m, 200m et 300m) pour les filaires N1 à N7.

On valide cette arborescence ? Si oui, dès que le fichier est en place, on passe à l'analyse spatiale par buffer.

Revision #4

Created 2026-03-03 10:06:10 UTC by Julien

Updated 2026-03-06 10:58:40 UTC by Julien