

Documentation Technique

Complétion des Données de Consommation Électrique Manquantes

Projet SolarX - Équipe de développement

April 4, 2025

Objectif

Certaines communes présentes dans le fichier GeoJSON de la carte n'ont pas de données de consommation électrique enregistrées dans la base de données. Cela entraîne un affichage partiel (zones grises) sur la carte de consommation. L'objectif est de combler ces lacunes avec une méthode simple, reproductible et cohérente.

1 Méthodologie

1.1 1. Extraction des communes du GeoJSON

Les noms des communes sont extraits depuis les propriétés `feature.properties.name` du fichier `geo_data_boundaries.geojson`.

```
geojson_communes = set(
    feature['properties']['name'].strip().lower()
    for feature in geojson_data['features']
    if 'name' in feature['properties'] and feature['properties']['name']
)
```

1.2 2. Récupération des communes déjà en base

Les communes ayant une consommation enregistrée en base de données sont récupérées via une requête SQL, puis converties en minuscules.

```
communes_bdd = set(df['nom_commune'].str.strip().str.lower())
```

1.3 3. Identification des communes manquantes

On calcule la différence entre les communes du GeoJSON et celles de la base de données pour trouver celles qui manquent.

```
communes_manquantes = geojson_communes - communes_bdd
```

1.4 4. Calcul de la moyenne globale de consommation

Une valeur moyenne globale de consommation électrique est calculée sur les données existantes.

```
moyenne_globale_conso = df['consommation'].mean()
```

1.5 5. Création des données manquantes

Un nouveau DataFrame est créé, assignant la moyenne globale à chaque commune manquante.

```
communes_manquantes_df = pd.DataFrame({
    'nom_commune': [commune.capitalize() for commune in communes_manquantes],
    'consommation': moyenne_globale_conso,
    'annee': 2023
})
```

1.6 6. Fusion avec les données existantes

Les données manquantes sont ajoutées au DataFrame principal.

```
df = pd.concat([df, communes_manquantes_df], ignore_index=True)
```

1.7 7. Mise à jour de la base de données

Chaque commune manquante est insérée dans la base avec la moyenne globale. Une mise à jour est effectuée si la commune existe déjà.

```
for _, row in communes_manquantes_df.iterrows():
    cursor.execute("""
        INSERT INTO 2026_solarx_consommation (nom_commune, consommation, annee)
        VALUES (%s, %s, %s)
        ON DUPLICATE KEY UPDATE consommation = VALUES(consommation)
    """, (row['nom_commune'], row['consommation'], row['annee']))
```

2 Avantages de l'approche

- Méthode simple, reproductible et rapide à exécuter.
- Aucun besoin de calculs complexes ou de modèles prédictifs.
- Toutes les communes visibles sur la carte disposent désormais d'une donnée de consommation cohérente.

Conclusion

Cette méthode assure une cohérence des données visuelles dans l'application et constitue une base solide pour de futures améliorations plus fines (par exemple : estimation par proximité géographique ou par corrélation météorologique).