



PROJET OPEN DATA PARIS - ARBRES REMARQUABLES

Objectif : Démontrer la capacité à récupérer, traiter et visualiser des données
Open Data

réalisé par : Yasmine Hedfi

OBJECTIF

- Récupérer des données réelles (API Open Data Paris)
- Les stocker et les structurer
- Les analyser et produire des indicateurs
- Les visualiser sur un dashboard simple

ARCHITECTURE ET FLUX DE DONNÉES :

API Paris ➔ PostgreSQL ➔ Backend Go ➔ Frontend Chart.js



REST API

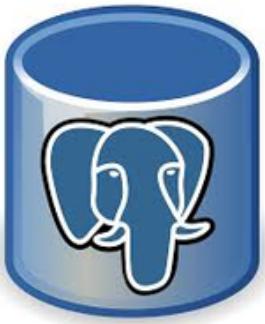


Chart.js

Étapes principales :

- 1. Récupération des données
- 2. Traitement et stockage
- 3. Création de l'API backend
- 4. Visualisation via frontend (Chart.js)

RÉCUPÉRATION DES DONNÉES :

- Utilisation de l'API Open Data Paris
- Données brutes au format JSON
- **Exemples** : identifiant, nom usuel, espèce, arrondissement, coordonnées...
Étape clé : transformer les données brutes en données exploitable

TRAITEMENT & STOCKAGE

- Stockage des données dans PostgreSQL
 - Création automatique de la table si besoin (idempotence)
 - Normalisation des champs : latitude, longitude, hauteur...
- >La base devient la « mémoire » fiable du projet

BACKEND EN GO

- API construite en Go
- Se connecte à PostgreSQL et expose les données
- Gestion des retries pour attendre que la DB soit prête

Rôle : rendre la donnée accessible au frontend

FRONTEND & VISUALISATION

- Petite page HTML + Chart.js
- Graphiques simples et parlants :
 - Nombre d'arbres par arrondissement
 - Hauteur moyenne par arrondissement
 - Répartition par genre
- Transformer les chiffres en visuels clairs

DÉPLOIEMENT :

- Dockerfile pour builder le backend
- docker-compose.yml pour orchestrer :
 - backend
 - db
 - frontend (serveur statique)
- Une seule commande : docker-compose up --build

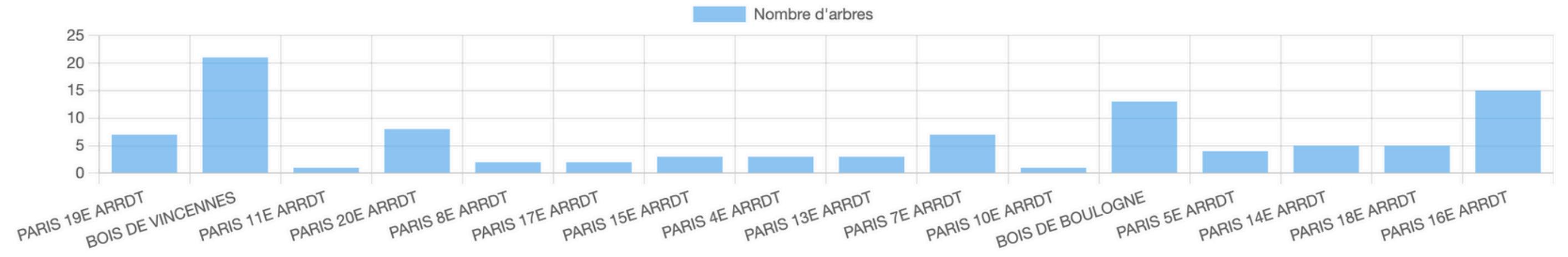
RESPECT DU 12 FACTOR APP :

- **Config** : tout est dans les variables d'environnement (ex. DATABASE_URL), rien de caché dans le code.
- **Dépendances** : Go gère ça avec go.mod et Docker isole bien chaque service.
- **Ports** : le backend s'ouvre sur 8081, exposé par Docker.
- **Stateless** : le backend ne garde rien, c'est la base qui stocke tout.
- **Même environnement** : que ce soit en local ou en prod, on lance le même docker-compose up.
- **Logs** : tout s'affiche dans la console → simple à suivre et à monitorer.

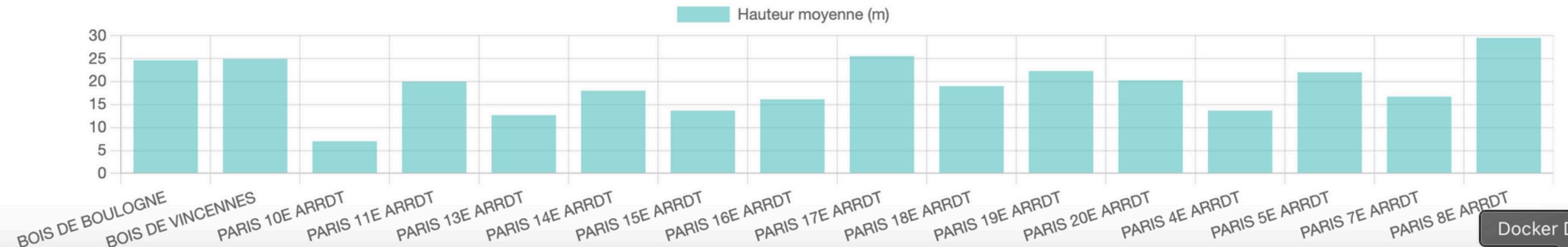
DEMO :

Arbres remarquables de Paris

Nombre d'arbres par arrondissement



Hauteur moyenne par arrondissement

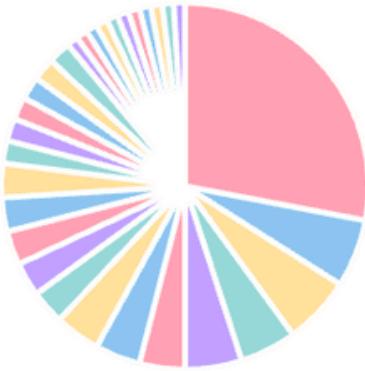


Docker Des

DEMO :

Répartition par genre

Platanus Fagus Quercus Cedrus Magnolia Diospyros Ginkgo Acer Sequoiadendron Aesculus Taxus
Pterocarya Corylus Styphnolobium Eucommia Pinus Liriodendron Ulmus Taxodium Robinia Davidia Maclura
Juglans Celtis Pistacia Ficus Catalpa Fraxinus Zelkova Araucaria



CONCLUSION

- Données récupérées et intégrées automatiquement depuis l'API Open Data Paris
- Traitement robuste avec Go + PostgreSQL
- Visualisation simple et efficace avec Chart.js
- Déploiement complet et reproductible avec Docker / Docker Compose
- Respect des bonnes pratiques 12 Factor App

THANK YOU