

TP 1 – Web Scraping & Stockage MongoDB : Données Météo

📌 Objectif

Ce TP vous apprend à :

- Scraper des données météo en temps réel depuis un site web
 - Les stocker dans une base de données MongoDB
 - Réaliser des requêtes pour exploiter ces données
-

📌 Stack utilisée

- Python (requests , BeautifulSoup , pymongo , dotenv)
 - MongoDB (local ou cloud)
 - Terminal + éditeur de code
-

📌 Données à utiliser

Site de scraping : <https://www.infoclimat.fr/observations-meteo/temps-reel/>

📌 Étapes

1. Scraping

Créer un script Python `scraper.py` qui récupère pour chaque station météo :

- Ville
- Température
- Pression
- Vent
- Heure de mise à jour

Données à retourner sous forme de liste de dictionnaires.

2. Insertion MongoDB

Créer un script `mongo_insert.py` qui :

- Se connecte à MongoDB (`MONGO_URI` à stocker dans un fichier `.env`)
- Insère les données scrapées dans une base `tp_meteo`, collection `stations`

3. Requêtes

Créer un script `requetes.py` qui permet de :

- Afficher les stations avec une température > 25°C
 - Calculer la température moyenne par ville (`$group`)
 - Trier les villes par température moyenne décroissante
 - Trouver la station avec la pression la plus élevée
-

📌 Bonus

- Lancer le scraping automatiquement toutes les heures (via `cron`)
 - Historiser les données dans Mongo (avec un champ `date`)
 - Visualiser les résultats dans un mini dashboard
-

TP 2 – Cas Pratique : Conception d’une architecture hybride NoSQL + SQL

📄 Contexte

Une entreprise veut analyser les actions de ses utilisateurs en ligne et sur ses bornes physiques. Elle collecte des logs en temps réel (clics, pages vues, actions...).

📄 Objectif du TP

Tu dois proposer une **architecture de stockage** adaptée pour :

- Recevoir et stocker les logs bruts
 - Les analyser rapidement
 - Les exploiter dans des outils métier
 - Les croiser avec les données clients déjà existantes dans une base SQL
-

📄 À rendre : un document PDF ou DOC avec

1. Où vont les données ?

▮ Où stocker les logs qui arrivent tout le temps et en grande quantité ? Quelle base NoSQL ? Pourquoi ce choix ?

📄 *Pense : JSON, gros volume, écriture rapide, pas de structure fixe → MongoDB, Elasticsearch ?*

2. Comment les exploiter ?

▮ Une fois stockées, que faire pour les analyser ? Peut-on les exporter vers du SQL ? Pourquoi ?

📄 *Pense : nettoyage, transformation en tableau, outils d'analyse, analystes métier...*

3. Architecture (schéma à dessiner)

▮ Fais un schéma simple montrant :

- où arrivent les données
- dans quelle base elles vont
- comment elles sont traitées
- où elles finissent pour l'analyse

📄 *Fais un dessin clair avec flèches →*

4. Performance & sécurité

▮ Comment garantir de bonnes performances ? Comment protéger les données ?

📄 *Pense : index, archivage, suppression auto, RGPD, chiffrement, droits d'accès*

📄 Bonus

- Exemple de statistique utile pour le métier
- Automatisation du transfert NoSQL → SQL (batch, stream...)