

Tableau de bord de suivi de l'avancement des indicateurs

Le principe est que l'avancement de la rédaction des fiches, les commentaires, les questionnements se voient à travers un tableur de données CSV -
"\\NEXTE\\Alteris\\Production\\AL_Inventaire_Structure.csv"

Il est mis en place par un script PYTHON généré par Gemini.
Ce script est évolutif - notamment pour y inserer les renvois vers les fiches Bookstack ou autre.
Ne pas changer de place le script (conçu en chemin relatif pour le NAS)- le tableur créé est lui mobile.

Extrait du tableur de suivi de l'avancement des indicateurs

Redimensionner le tableau		Convertir en plage		segment	Supprimer la liaison		☒ Lignes à bandes		☐ Colonnes à bandes		
Propriétés		Outils		Données de tableau externe		Options de style de tableau					
A1	⌵	:	⌵	fx							
	A	B	C	D	E	F	G	H			
1	Column1	Column2	Column3	Column4	Column5						
2	THEMATIQUE	NATURE	ID_SQL	NOM_INDICATEUR	STATUT						
3	CHANGEMENT_CLIMATIQUE	Impact	I_T3_01	Artificialisation des sols	A rédiger						
4	CHANGEMENT_CLIMATIQUE	Impact	I_T3_02	Reduction des surfaces vegetalisees	A rédiger						
5	CHANGEMENT_CLIMATIQUE	Impact	I_T3_03	Ilots de chaleur urbains	A rédiger						
6	CHANGEMENT_CLIMATIQUE	Impact	I_T3_04	Impermeabilisation excessive	A rédiger						
7	CHANGEMENT_CLIMATIQUE	Impact	I_T3_05	Fragilite des digues et protections hydrauliques	A rédiger						
8	CHANGEMENT_CLIMATIQUE	Impact	I_T3_06	Urbanisation en zone a risque	A rédiger						
9	CHANGEMENT_CLIMATIQUE	Impact	I_T3_07	Vulnerabilite des reseaux d'eau	A rédiger						
10	CHANGEMENT_CLIMATIQUE	Impact	I_T3_08	Vieillessement du patrimoine bati	A rédiger						
11	CHANGEMENT_CLIMATIQUE	Impact	I_T3_09	Insuffisance des refuges climatiques urbains	A rédiger						
12	CHANGEMENT_CLIMATIQUE	Impact	I_T3_10	Dependance a des ressources hydriques fragiles	A rédiger						
13	CHANGEMENT_CLIMATIQUE	Impact	I_T3_11	Fragilite des forets periurbaines	A rédiger						
14	CHANGEMENT_CLIMATIQUE	Impact	I_T3_12	Absence de planification de resilience climatique	A rédiger						
15	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_01	SYM_Dégradation des espaces comuns	A rédiger						
16	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_02	SYM_Multiplication des ilots de chaleur	A rédiger						
17	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_03	SYM_Anxieté climatique	A rédiger						
18	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_04	SYM_Inondations recurrentes	A rédiger						
19	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_05	SYM_Fragilisation des écosystèmes	A rédiger						
20	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_06	SYM_Stresse hydrique	A rédiger						
21	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_07	SYM_Glissements de terrain	A rédiger						
22	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_08	SYM_Incendies et evacuations	A rédiger						
23	CHANGEMENT_CLIMATIQUE	Symptome	S_T3_09	SYM_Sécheresse	A rédiger						
24											
25	CRISE_DE_CONFIANCE	Impact	I_T1_01	Fragmentation de l'espace public informationnel	A rédiger						
26	CRISE_DE_CONFIANCE	Impact	I_T1_02	Mefiance envers l'expertise scientifique	A rédiger						
27	CRISE_DE_CONFIANCE	Impact	I_T1_03	Sentiment d'injustice territoriale	A rédiger						

le script évolutif

```
import os
import csv
import re

# --- CONFIGURATION DYNAMIQUE ---
# Le script regarde là où il est enregistré
script_dir = os.path.dirname(os.path.abspath(__file__))

# On définit le nom du dossier Obsidian (qui doit être dans le même dossier que le script)
folder_name = "Obsidian_Alteris"
base_dir = os.path.join(script_dir, folder_name)

# Le CSV sera créé au même endroit que le script
output_file = os.path.join(script_dir, "AL_Inventaire_Structure.csv")

data_inventory = []

print(f"--- PILOTAGE ALTERIS ---")
print(f"Recherche du coffre dans : {base_dir}")

if not os.path.exists(base_dir):
    print(f"ERREUR : Le dossier '{folder_name}' est introuvable.")
    print(f"Vérifiez que ce script est bien placé à côté du dossier du NAS.")
else:
    # ... (Le reste du code reste identique pour l'extraction et le tri) ...
    for root, dirs, files in os.walk(base_dir):
        for file in files:
            if file.endswith(".md") and not file.startswith("00_"):
                file_path = os.path.join(root, file)
                try:
                    with open(file_path, "r", encoding="utf-8") as f:
                        content = f.read()
                        id_sql = re.search(r"ID_SQL:\s*(.*)", content)
                        nature = re.search(r"Nature:\s*(.*)", content)
                        statut = re.search(r"Statut:\s*(.*)", content)
```

```

id_val = id_sql.group(1).strip() if id_sql else "999"
nature_val = nature.group(1).strip() if nature else "Inconnue"

parts = file_path.split(os.sep)
thematique = "Inconnue"
if "Impacts" in parts:
    thematique = parts[parts.index("Impacts")-1]
elif "Symptomes" in parts:
    thematique = parts[parts.index("Symptomes")-1]

data_inventory.append({
    "THEMATIQUE": thematique,
    "NATURE": nature_val,
    "ID_SQL": id_val,
    "NOM_INDICATEUR": file.replace(".md", ""),
    "STATUT": statut.group(1).strip() if statut else "A rédiger"
})
except Exception as e:
    pass

# Tri et écriture
data_inventory.sort(key=lambda x: (x['THEMATIQUE'], x['NATURE'], x['ID_SQL']))

if data_inventory:
    with open(output_file, "w", encoding="utf-8-sig", newline="") as csvfile:
        fieldnames = ["THEMATIQUE", "NATURE", "ID_SQL", "NOM_INDICATEUR", "STATUT"]
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames, delimiter=";")
        writer.writeheader()
        current_thematique = ""
        for row in data_inventory:
            if current_thematique != "" and row["THEMATIQUE"] != current_thematique:
                writer.writerow({k: "" for k in fieldnames})
            writer.writerow(row)
            current_thematique = row["THEMATIQUE"]
    print(f"SUCCÈS ! Inventaire mis à jour dans ce dossier.")

```