

```
import csv
import json
import subprocess
import os

# Fonction pour rechercher le fichier comorbidites.json dans plusieurs emplacements courants
def find_comorbidites_file():
    common_locations = [
        os.path.join(os.path.expanduser('~'), 'Desktop', 'comorbidites.json'),
        os.path.join(os.path.expanduser('~'), 'Downloads', 'comorbidites.json'),
        # Ajoutez d'autres emplacements au besoin
    ]

    for location in common_locations:
        if os.path.isfile(location):
            return location

    return None

# Recherche du fichier comorbidites.json dans plusieurs emplacements courants
file_path = find_comorbidites_file()

if file_path:
    # Charger les données JSON depuis le fichier
    with open(file_path, 'r', encoding='utf-8') as file:
        data_list = json.load(file)

    csv_file = "output.csv"

    # Vérifier que la liste contient des dictionnaires avec un champ "fields"
    if all(isinstance(entry, dict) and "fields" in entry for entry in data_list):
        # Définir les noms de colonnes que vous souhaitez extraire
        selected_columns = ["patho_niv1", "libelle_comorbidite", "annee", "ntop", "ncomorb", "proportion_comorb", "niveau_prioritaire"]

        # Trier les données par plusieurs colonnes
        sorted_data_list = sorted(data_list, key=lambda x: (
            x.get("fields", {}).get("proportion_comorb", 0),
            x.get("fields", {}).get("patho_niv1", ""),
            x.get("fields", {}).get("libelle_comorbidite", ""),
            x.get("fields", {}).get("annee", "")
        ), reverse=True)

        # Traitement de la colonne "niveau_prioritaire"
        for entry in sorted_data_list:
            if "fields" in entry:
                niveau_prioritaire_values = entry["fields"].get("niveau_prioritaire", [])
                if niveau_prioritaire_values:
                    # Remplacer la colonne "niveau_prioritaire" par la valeur la plus élevée
                    max_niveau_prioritaire = max(niveau_prioritaire_values)
                    entry["fields"]["niveau_prioritaire"] = max_niveau_prioritaire

        # Écriture des données triées dans le fichier CSV avec un point-virgule comme délimiteur
        with open(csv_file, 'w', newline='', encoding='utf-8-sig') as file:
            # Utilisez DictWriter en spécifiant le délimiteur comme point-virgule
            csv_writer = csv.DictWriter(file, fieldnames=selected_columns, delimiter=';')

            # Écrire l'en-tête du fichier CSV
            csv_writer.writeheader()

            # Écrire les données triées des colonnes sélectionnées dans le fichier CSV
            for entry in sorted_data_list:
                if "fields" in entry:
                    # Vérifier si toutes les colonnes sélectionnées ont des valeurs non vides
                    if all(entry["fields"].get(key, "") for key in selected_columns):
                        # Arrondir la valeur de la colonne "proportion_comorb" à trois chiffres après la virgule
                        entry["fields"]["proportion_comorb"] = round(entry["fields"]["proportion_comorb"], 3)

                        # Créer un dictionnaire avec seulement les colonnes sélectionnées
                        selected_data = {key: entry["fields"].get(key, "") for key in selected_columns}
                        csv_writer.writerow(selected_data)

        print(f"Le fichier CSV '{csv_file}' a été créé avec succès")

        # Ouvrir le fichier CSV avec l'application par défaut
        subprocess.run(['start', 'excel', csv_file], shell=True)
    else:
        print("La liste ne contient pas que des dictionnaires avec un champ 'fields'.")
else:
    print("Erreur : Le fichier comorbidites.json n'a pas été trouvé dans les emplacements courants.")
```