

## Piwigo et uMap

On m'a transmis un script en python (*généré avec ChatGPT*) pour récupérer les données GPS de photos hébergées sur un serveur Piwigo pour les positionner sur une carte créée avec uMap. J'ai adapté ce script au Debian 11 de mon serveur Piwigo.

### Nouvelle version

La version précédente du script génère un fichier qui ne met pas en place **les entêtes** et contient les **messages d'erreur**. Voici donc une nouvelle version `umap2.py` qui corrige cela. le fichier créé est **umap.csv**, s'il existe déjà il est **automatiquement remplacé**.

J'utilise le **SSH** avec **PuTTY** sous Windows pour prendre la main en lignes de commande sur le serveur Piwigo qui héberge les photos (dans mon cas je peux aussi utiliser la console de mon hyperviseur Promox). Pour transférer le script Python dans mon dossier utilisateur sur le serveur Piwigo j'ai utilisé **FileZilla**.

### Prérequis

**Le script doit être installé et exécuté sur le serveur Piwigo.**

**Le serveur SSH a été installé sur la machine qui héberge Piwigo lors de l'installation de Debian.**

Tout d'abord sur mon serveur Piwigo j'ai regardé quelle est la version de Python 3 installée :

```
python3 --version  
Python 3.9.2
```

Installation des paquets nécessaires à la mise en place des bibliothèques pour Python 3, en administrateur (root pour Debian ou sudo pour Ubuntu, Mint). Pip est un gestionnaire de paquets :

```
apt install python3-pip
```

Installation de la bibliothèque pillow pour Python 3 avec pip (équivalent à PIL) :

```
pip3 install pillow
```

### Le script en Python

Mon script `umap2.py` modifié :

---

```
#!/usr/bin/env python3.9  
  
#-*- coding: utf-8 -*-  
  
from PIL import Image  
from PIL.ExifTags import TAGS  
import os  
  
def convert_gps_to_decimal(gps_data):  
    lat_deg, lat_min, lat_sec = gps_data[2]  
    lon_deg, lon_min, lon_sec = gps_data[4]  
  
    lat = lat_deg + lat_min / 60.0 + lat_sec / 3600.0  
    lon = lon_deg + lon_min / 60.0 + lon_sec / 3600.0
```

```

# Ajuster en fonction de la direction (N, S, E, W)
lat = lat if gps_data[1] == 'N' else -lat
lon = lon if gps_data[3] == 'E' else -lon

return lat, lon

def create_image_url(chemin_base, fichier):
    # Ajouter le suffixe "-xx" juste avant l'extension .jpg
    nom_fichier, extension = os.path.splitext(fichier)
    fichier_avec_suffixe = f"{nom_fichier}-xx{extension}"

    # Construire l'URL complète avec le nom du fichier
    image_url = f"{{{http://{chemin_base}/{fichier_avec_suffixe}}}}}"
    return image_url

def extract_and_convert_gps_info(chemin_photos, chemin_base):
    if os.path.exists("./umap.csv"):
        os.remove("./umap.csv")
        print("fichier umap.csv déjà existant supprimé")
    # a, pour une ouverture en mode ajout à la fin du fichier (APPEND). Si le
    fichier n'existe pas python le crée.
    global fichier_csv
    fichier_csv = open("umap.csv", "a")
    fichier_csv.write("URL,Lat,Lon")
    for fichier in os.listdir(chemin_photos):
        chemin_complet = os.path.join(chemin_photos, fichier)

        try:
            with Image.open(chemin_complet) as img:
                exif_data = img._getexif()

                if exif_data is not None:
                    # Extraire les données GPS
                    gps_info = {TAGS[key]: exif_data[key] for key in
exif_data.keys() if key in TAGS and TAGS[key] == 'GPSInfo'}

                    if gps_info:
                        # Convertir les coordonnées GPS en décimal
                        coordinates_decimal =
convert_gps_to_decimal(gps_info['GPSInfo'])
                        if coordinates_decimal:
                            # Créer l'URL avec le nom du fichier
                            image_url = create_image_url(chemin_base, fichier)
                            # Imprimer le résultat sous forme de chaîne
                            result_str = f"{image_url},{coordinates_decimal[0]},
{coordinates_decimal[1]}"
                            print(result_str)
                            # saut de ligne \n
                            fichier_csv.write("\n" + result_str)

            except Exception as e:
                print(f"Erreur lors de l'extraction des données Exif pour
{chemin_complet}: {str(e)}")

# Chemin du répertoire contenant les photos
chemin_photos = r'/var/www/html/piwigomv/upload/2023/05/06'

# Chemin de base pour la création de l'URL
chemin_base = 'piwigo.memoirevive79.mooo.com/i.php?/upload/2023/05/06'

```

```
# Appeler la fonction pour extraire et convertir les informations GPS
extract_and_convert_gps_info(chemin_photos, chemin_base)
fichier_csv.close()
print("fichier umap.csv créé")
```

---

Pour Debian j'ai ajouté les lignes suivantes au début du script :

```
#!/usr/bin/env python3.9
```

```
#!/usr/bin coding: utf-8 -*-
```

*J'ai vérifié la version de python au début.*

On doit adapter le chemin\_photos et le chemin\_base mes photos se trouvent dans un sous dossier d'upload. J'ai conservé i.php?.

Pour que mon fichier Python .py soit exécutable, depuis le terminal :

```
chmod +x umap2.py
```

Pour exécuter mon fichier Python depuis le dossier où il se trouve dans un terminal :

```
./umap2.py
```

On peut récupérer le fichier umap.csv avec **FileZilla**.

---

## Le fichier umap.csv

Voilà les données obtenues dans LibreOffice Calc. On voit les entêtes URL, Lat, Lon

	A	B	C
1	URL	Lat	Lon
2	{{http://piwigo.memoirevive79.mooc.com/i.php?/upload/2023/05/06/20230506160707-bbb85c3e-xx.jpg}}	46.31916666666667	-0.3444444444444444
3	{{http://piwigo.memoirevive79.mooc.com/i.php?/upload/2023/05/06/20230506160711-26b7ae40-xx.jpg}}	46.31027777777778	-0.33388888888888885
4	{{http://piwigo.memoirevive79.mooc.com/i.php?/upload/2023/05/06/20230506160706-fcae0807-xx.jpg}}	46.31916666666667	-0.34388888888888886
5	{{http://piwigo.memoirevive79.mooc.com/i.php?/upload/2023/05/06/20230506160702-57b91b7f-xx.jpg}}	46.31916666666667	-0.34388888888888886
6	{{http://piwigo.memoirevive79.mooc.com/i.php?/upload/2023/05/06/20230506160713-4e3ce67f-xx.jpg}}	46.335	-0.4122222222222227
7			
8			

## Sur le site uMap

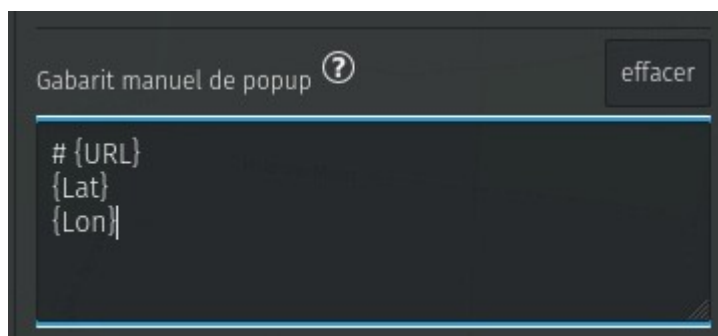
Sur le site <https://umap.openstreetmap.fr/>

## Pour importer mon fichier csv

**Importer des données (CTRL + I) ==> [Parcourir...] ==> Sélectionnez votre fichier ==> [Importer]**

## Pour voir les photos

Dans « **Éditer les propriétés de la carte** » (la roue dentée) ==> **Options d'interaction par défaut** ==> **Gabarit manuel de popup**



J'ai saisi les paramètres correspondant à mes entêtes en respectant la casse.

En fait dans mon cas il semble que seul {URL} soit utile.

## Pour le diaporama

Dans « **Éditer les propriétés de la carte** » (la roue dentée) ==> **Diaporama** ==> **Activer le mode diaporama sur ON**

Pour changer la forme de l'icône pour l'épingle

Dans « **Éditer les propriétés de la carte** » (la roue dentée) ==> **Propriétés de forme par défaut** ==> **Forme de l'icône sur Épingle**

Pour agrandir les photos

Dans « **Éditer les propriétés de la carte** » (la roue dentée) ==> **Options d'interaction par défaut** ==> **Forme de popup sur Popup (grande)**

Pour définir les limites géographiques à la vue courante

Dans « **Éditer les propriétés de la carte** » (la roue dentée) ==> **Limites géographiques** ==> **[Utiliser la vue courante]**

## Le résultat obtenu

[https://umap.openstreetmap.fr/fr/map/vouille-gascougnolles\\_1028350](https://umap.openstreetmap.fr/fr/map/vouille-gascougnolles_1028350)

Vous pouvez cliquer sur le bouton « **Commencer** ».

---

## Pour insérer la carte dans une page ou un article d'un site WordPress

Dans Umap cliquez sur l'icône **Partager et télécharger** à gauche, copier le code qui se trouve sous **Intégrer la carte dans iframe**.

Dans votre page ou article WordPress cliquez sur les **trois points** en haut à droite, dans la section Éditeur cochez **Éditeur de code**, vous pouvez ensuite coller votre code. Vous pouvez ensuite revenir à l'**Éditeur visuel**.

---

## Sublime Text

Sous Debian ou Linux Mint, pour éditer mon script Python je peux utiliser Sublime Text :

[https://www.sublimetext.com/docs/linux\\_repositories.html](https://www.sublimetext.com/docs/linux_repositories.html)

Voir l'installation pour apt (*avec Debian en administrateur root pas de commande sudo*).

Configuration de Sublime Text pour Python :

Depuis le menu **View ==> Syntax**, sélectionnez **Python**.

---

## Accès SSH sous Linux

Pour accéder au serveur distant Piwigo depuis un terminal de mon poste de travail sous Linux :

```
ssh utilisateur@IP_public -p port
```

- **utilisateur** : le nom de mon utilisateur sur le serveur Piwigo.
- **IP\_public** : Adresse IP publique du site distant où se trouve le serveur Piwigo. Adresse IP du serveur si l'on est sur le même réseau local.
- **port** : Le port SSH à utiliser, si on ne précise pas ce paramètre on utilisera le port par défaut qui est le 22.

Pour quitter :

```
exit
```

Plus d'informations

[Programmer en Python sous Debian](#)

---

Dominique Renaudeau - 24/02/2024