

# Collecte et centralisation des données des maquettes Revit

## I. Contexte

Le développement de ce script Python répond à une demande spécifique du BIM Manager de l'agence. Face à la complexité croissante de la gestion des projets et à la multiplication des maquettes Revit, il est devenu essentiel de centraliser et de simplifier l'accès aux informations clés de ces maquettes. L'objectif est de fournir au BIM Manager une vision globale et consolidée de l'ensemble des maquettes Revit stockées sur le serveur de fichiers, facilitant ainsi le suivi des projets, l'extraction de données pertinentes et la prise de décisions éclairées.

Ce script vise à automatiser la collecte des informations essentielles (nom du fichier, numéro d'affaire, nom de l'affaire, taille du fichier, chemin d'accès, date de dernière modification) et à les rendre facilement accessibles via un fichier Google Sheets. Cette approche permettra au BIM Manager de disposer d'un outil simple et efficace pour visualiser l'état d'avancement des projets, identifier les maquettes les plus volumineuses, suivre les modifications et garantir la conformité des données.

## II. Prérequis

- Python installé (version 3.8 minimum recommandée)
- Accès au serveur FTP et au serveur de fichier des projets
- Un fichier .env correctement configuré
- Bibliothèques nécessaires :
  - os
  - pandas
  - datetime
  - csv
  - ftplib
  - dotenv
  - logging

### III. Structure du Code

#### A. Chargement des Variables d'Environnement

```
dotenv_path = r'..\accessftp.env'|
load_dotenv(dotenv_path)

log_filename = r'..\ftp_upload.log'
logging.basicConfig(filename=log_filename, level=logging.ERROR, format='%(asctime)s - %(levelname)s - %(message)s')

# Récupérer les identifiants à partir des variables d'environnement
ftp_server = os.getenv('FTP_SERVER')
ftp_username = os.getenv('FTP_USERNAME')
ftp_password = os.getenv('FTP_PASSWORD')
ftp_directory = os.getenv('FTP_DIRECTORY')
```

- Utilisation de dotenv pour sécuriser les accès FTP.
- Fichier .env pour éviter le stockage en clair des identifiants dans le code.

#### B. Définition de la Recherche de Dossiers

```
def find_revit_folder(root_directory, exclude_projects=None, max_depth=3): 1 usage
    folders = []
    for root, dirs, files in os.walk(root_directory):
        # Calculer la profondeur actuelle
        current_depth = root.count(os.sep) - root_directory.count(os.sep)

        # Exclure les projets dans la liste d'exclusion
        project_name = os.path.basename(root)
        if project_name in exclude_projects:
            dirs.clear()
            continue

        # Variable pour indiquer si un dossier a été trouvé dans ce projet
        folder_found = False

        if current_depth <= max_depth:
            if '00-GENERAL' in dirs:
                revit_path = os.path.join(root, '00-GENERAL')
                folders.append(revit_path)
                folder_found = True
                # Faire quelque chose avec revit_path trouvé
```

- Fonction find\_revit\_folder :

- Explore récursivement un répertoire parent (profondeur max de 3 niveaux).
- Exclut des projets spécifiques prédéfinis.
- Ne conserve que les chemins contenant un dossier OO-GENERAL.

## C. Extraction d'Informations sur les Fichiers

```
def determine_revit_version(maquette): 1 usage
    version = None
    current_year = datetime.now().year
    for i in range(10, current_year - 2000 + 1): # Vérifie de 2010 à l'année actuelle
        year_suffix = str(i).zfill(2) # Assure que le suffixe à deux chiffres
        full_year = '20' + year_suffix
        if any(keyword in maquette for keyword in [f'-20{year_suffix}', f'R{year_suffix}', f'_A{year_suffix}', f'-A{year_suffix}']):
            version = full_year
    return version
```

- Fonction determine\_revit\_version :
  - Déduit la version Revit d'après des motifs trouvés dans le nom du fichier.

```
def get_maquette_info (maquette): 1 usage
    try:
        maquette_size = os.path.getsize(maquette) / (1024 * 1024) # taille en MB
        modification = os.path.getmtime(maquette)
        modification_date = datetime.fromtimestamp(modification).strftime('%Y-%m-%d')
        # Convertir le timestamp en un objet datetime et formatage de la date
        return maquette_size, modification_date
    except FileNotFoundError:
        return None, None
```

- Fonction get\_maquette\_info :
  - Récupère la taille (en MB) et la date de dernière modification d'un fichier .rvt.

## D. Construction du DataFrame

- Stockage des informations collectées dans un dictionnaire structuré.
- Transformation en DataFrame pandas.
- Conversion du numéro d'affaire en type entier pour faciliter le tri.
- Tri croissant par numéro d'affaire.

## E. Exportation en CSV

```
def export_to_csv(data, filename): 1 usage
    # Obtenir les en-têtes des colonnes à partir des clés du dictionnaire
    headers = list(data.keys())

    # Transposer le dictionnaire pour obtenir les lignes
    rows = zip(*data.values())

    with open(filename, 'w', newline='') as file:
        writer = csv.writer(file, delimiter=';')
        writer.writerow(headers) # Écrire les en-têtes
        writer.writerows(rows)  # Écrire les lignes de données
```

- Fonction `export_to_csv` :
  - Écrit les données dans un fichier CSV (séparateur ;).
  - Format facile à lire et réutilisable.

## F. Envoi sur FTP

```
def upload_to_ftp(filename, ftp_server, ftp_username, ftp_password, ftp_directory): 1 usage
    try:
        ftp = FTP(ftp_server)
        ftp.login(ftp_username, ftp_password)
        ftp.cwd(ftp_directory)

        with open(filename, 'rb') as file:
            ftp.storbinary(cmd=f'STOR {filename}', file)

        ftp.quit()
        print(f"Le fichier {filename} a été téléchargé avec succès sur le serveur FTP.")
    except Exception as e:
        error_message = f"Une erreur s'est produite lors du téléchargement FTP : {e}"
        print(error_message)
        logging.error(error_message)
```

- Fonction `upload_to_ftp` :
  - Connexion au serveur FTP avec authentification.
  - Navigation dans le bon répertoire.
  - Upload du fichier CSV.

## IV. Remarques Techniques

- Gestion des erreurs :
  - Le script logue les erreurs FTP dans un fichier .log.
- Attention aux chemins relatifs :
  - Bien utiliser des chemins relatifs ou absolus Windows (r!\chemin\...') et éviter les chemins UNC type \\serveur\....
- Robustesse :
  - Exclusion automatique de projets non pertinents.
  - Protection contre les erreurs de fichier manquant (FileNotFoundException).

## V. Évolutions Possibles

- Ajouter une interface CLI (command-line) pour passer dynamiquement :
  - Le chemin parent
  - La profondeur de recherche
  - Le fichier de sortie
- Supporter d'autres formats de sortie : JSON, Excel.
- Ajouter un test de connexion FTP avant upload pour plus de robustesse.
- Meilleure détection de version Revit basée sur métadonnées (si disponible).