

Introduction :

Ce dossier contient un ensemble de fichiers Python constituant une application permettant de créer des bibliothèques de livres aux formats EPUB et PDF. Cette application a la capacité de récupérer des livres présents sur internet par une méthode de scrapping et de les mettre dans la bibliothèque choisie. Il peut aussi générer des rapports aux formats EPUB et PDF de tous les livres présents dans une bibliothèque, c'est-à-dire des listes de chaque livre avec le titre, l'auteur, le format et le nom du fichier ordonnées soit par titre soit par auteur. Le fichier à exécuter est le fichier Bibli.py. Pour savoir comment utiliser le logiciel, veuillez ouvrir le fichier Notice.pdf. Les fichiers Python sont composés de scripts de différentes classes et fonctions utilisées, ainsi que des programmes permettant d'extraire les informations rentrées par l'utilisateur (dans le terminal et dans les fichiers .conf).

Modules complémentaires utilisés :

Le programme a besoin de modules qui ne sont pas déjà préinstallés dans Anaconda. Ce sont les modules :

- ebooklib (importé dans les fichiers fonctions_bibli.py et livre_epub.py)
- PyPDF2 (livre_pdf.py)
- lxml2pdf (fonctions_bibli.py)
- BeautifulSoup4 (fonctions_bibli.py)

Pour chacun de ces modules, en cas d'oubli d'installation, dans le fichier l'utilisant il y a un programme qui l'importe s'il est déjà installé et sinon l'installe automatiquement (sous forme de boucle try) et l'importe ensuite.

Contenu de chaque fichier :

Chaque fichier contient seulement la classe qui porte le même nom (par exemple le fichier base_bibli.py contient seulement la classe base_bibli) à l'exception des suivants :

- **Bibli.py** : C'est le fichier à exécuter. Il contient un programme permettant d'extraire la majorité des données rentrées par l'utilisateur puis d'exécuter les instructions pour créer la bibliothèque et télécharger les livres ou pour créer les quatre rapports. Pour cela il importe les classes définies dans les autres fichiers python.
- **Cbibli.py** : Contient la classe bibli.
- **fonctions_bibli.py** : Contient un ensemble de fonctions qui sont utilisées dans les autres fichiers python. ATTENTION : Ce fichier contient une variable globale VerificationSSL qui doit prendre la valeur False uniquement lorsqu'on compte faire un scrapping sur un site de confiance.
- **Bibli.conf** : C'est un fichier de configuration que l'utilisateur peut modifier (en suivant bien les règles établies dans la notice) pour changer certaines options lors du scrapping et de la création des rapports. Les paramètres dedans sont utilisés dans les fichiers Bibli.py et simple_bibli.py. Les programmes gèrent aussi bien le cas où l'utilisateur rentre l'argument -c et le nom d'un fichier de configuration personnalisé.

Rôle de chaque classe :

- **base_livre** : Cette classe abstraite est la base pour représenter n'importe quel livre. Elle contient des moyens simples pour obtenir des informations basiques sur un livre comme son titre, son auteur, etc.
- **base_bibli** : Une classe abstraite pour gérer une bibliothèque (c'est-à-dire une collection de livre). Elle permet d'ajouter des livres à la bibliothèque et de créer des rapports sur les livres qu'elle contient.
- **livre_epub** : Une sous-classe de **base_livre** pour manipuler les livres au format EPUB. Elle aide à obtenir des détails comme le titre, l'auteur et autres informations spécifiques sur les livres au format EPUB.
- **livre_pdf** : Une sous-classe de **base_livre** pour travailler avec des livres au format PDF. De même elle extrait les métadonnées comme le titre et l'auteur de ces livres.
- **simple_bibli** : Une classe pour créer une bibliothèque basique. Elle permet d'ajouter un livre à la bibliothèque depuis un chemin d'accès ou un lien (en le téléchargeant) et de générer des rapports sur tous les livres ou les auteurs présents. Une instance de classe **simple_bibli** est caractérisée uniquement par le chemin d'accès local rentré en argument lors de sa création et par les méthodes qu'on peut lui appliquer.
- **bibli** : Une sous classe de **simple_bibli**. Elle contient en plus la méthode `alimenter` qui ajoute tous les livres référencés dans la page web correspondant à l'URL rentrée en argument.
- **bibli_scrap** : Une sous classe de **bibli**. Elle ajoute la méthode `scrap` qui permet de récupérer des livres à partir d'une URL par scrapping, avec des options pour limiter le nombre de livres récupérés et la profondeur de la recherche, c'est-à-dire le nombre d'URL visitées.

Liste des modifications faites par rapport au travail rendu à l'étape I :

- Tout le code est maintenant divisé en plusieurs fichiers.
- Mise en place des programmes pour extraire les arguments rentrés dans le terminal et les fichiers de configuration comme `bibli.conf` (présents dans les fichiers `Bibli.py` et `simple_bibli.py`).
- Les boucles `try` pour installer les modules s'ils ne sont pas déjà installés.
- L'ajout de la vérification « `req.status_code==200` » avant d'interagir avec une URL.
- Le remplacement de la fonction `nomfichier()` par la fonction prédéfinie `osp.basename()`.
- Le message d'erreur "Pas de livre dans la bibliothèque !" si on cherche à générer le rapport d'une bibliothèque vide.
- Les messages d'affichage progressifs ("Rapport_des_auteurs.epub créé.") et finaux ("Génération des 2x2 rapports effectuée avec succès.") lors de la génération des rapports.

- Le message d'erreur "Profondeur entrée incohérente. Veuillez..." lorsque l'utilisateur rentre une profondeur de recherche qui n'est pas un entier strictement positif lorsqu'il demande un scrapping.
- La fin de l'affichage de toutes les alertes (celles du module ebooklib et celles du au fait qu'on a désactivé la vérification du SSH).
- Modification de la fonction Download : renvoie True si le téléchargement a marché et False sinon, et gère les exceptions.
- Modification de la méthode ajouter : renvoie True si le téléchargement a marché si le fichier était en ligne. Note : Ces deux dernières modifications n'ont pas pu être vérifiées que sur Windows donc en cas de problème utilisez les versions mis en commentaire de la fonction Download et de la partie dans la méthode ajouter.
- L'affichage du nombre de livres téléchargés tous les 50 livres et du nombre de sites visités tous les 10.
- L'affichage de l'un des 3 messages finaux possibles à la fin du scrapping, indiquant le nombre total de livres téléchargés et de sites visités.
- Note : De même, le fichier bibli_scrap.py n'a pas été vérifié sur Linux mais uniquement sur Windows. En cas de problème d'incompatibilité, remplacer-le par le fichier présent dans le dossier bibli_scrap_L qui lui a été vérifié sur Linux, et remplacer aussi tous les autres codes qui n'ont pas été vérifiés sur Linux comme expliquer plus haut en note.