

Documentation

Le code fournit est un programme python orienté objet qui implémente une gestion de bibliothèque de livres électroniques (pdf, epub). Il est organisé en plusieurs classes pour représenter différents types de livres et bibliothèques, et il propose des fonctionnalités telles que la création de rapports sur les livres et l'alimentation de la bibliothèque à partir de ressources en ligne (web scraping). Ci-après une documentation explicative du code.

D'abord il nous faudra importer quelques bibliothèques :

- `from reportlab.pdfgen import canvas` : pour écrire du texte, éditer dans un fichier pdf.
- `from ebooklib import epub` : pour écrire et lire les fichiers de type epub.
- `import PyPDF2` : pour récupérer les metadatas d'un fichier pdf.
- `from bs4 import BeautifulSoup` : pour scraper (naviguer, rechercher) dans les fichiers HTML ou XML.
- `import requests` : permet d'utiliser le protocole http de façon simple.
- `from urllib.parse import urlparse, urljoin` : pour la gestion des url.
- `import os` : fournit une façon portable d'utiliser les fonctionnalités dépendantes du système d'exploitation.

➤ `class base_livre` : Cette classe permet de modéliser une livre. Nous définissons également les sous classes `LivreEPUB` et `LivrePDF` qui héritent de cette dernière, car nous travaillons avec deux types de formats pdf et epub. Elles nous permettront d'extraire à partir de la ressource du livre (nom ou url) des informations telles que le titre du livre, l'auteur, la date de publication ...

➤ `class base_bibli` : Cette classe nous permet de modéliser une bibliothèque grâce au répertoire contenant les livres de cette bibliothèque. Nous définissons la sous classe `SimpleBibli` dans laquelle on a : la méthode (`ajouter`) qui sert à ajouter un livre à la

bibliothèque, la méthode `rapport_livres` qui va nous permettre de générer un rapport soit en format pdf ou epub (en fonction du format désiré par l'utilisateur) des livres de la bibliothèque. Dans ce rapport, pour chaque livre on pourra voir son titre, son auteur, et son type (pdf ou epub). La méthode `rapport_auteurs` qui renvoie un rapport des auteurs en format pdf ou epub. Dans ce rapport, pour chaque auteur on a le titre de ses livres en bibliothèque et le type de chacun de ses livres.

- class `Bibli` : hérite de `SimpleBibli` et elle a en plus une méthode (`telecharger`) qui va nous permettre de télécharger des livres à partir de leurs ressources. Aussi, elle a une méthode (`alimenter`) qui appelle la méthode (`telecharger`) par laquelle on pourra télécharger les livres contenus dans l'url passé par l'utilisateur avant de les insérer dans la bibliothèque.
- class `Bibli_scrap` : hérite de `Bibli` et possède une méthode (`scap`). La méthode `scrap` permet d'alimenter la bibliothèque de type `Bibli_scrap` en respectant deux conditions d'arrêt imposées par l'utilisateur : la profondeur qui est le nombre maximal de page web à traiter et `nbmax` qui le nombre maximal de livres à télécharger. Cette méthode utilise la même démarche que la méthode `alimenter` mais en décrémentant à chaque fois le paramètre `nbmax` jusqu'à ce qu'il atteigne 0. Ensuite cette méthode s'appelle elle-même récursivement mais en décrémentant la profondeur. Ce processus se répète jusqu'à atteindre la condition d'arrêt qui est l'atteinte du nombre 0 par l'un des paramètres.

NB : Tout le long du code, nous avons fait en sorte de gérer certaines exceptions afin de garantir la robustesse du programme. Notamment les échecs de téléchargements de certains livres, les formats non supportés par l'application (autre que epub et pdf), les fichiers corrompus ...