

Université d'Angers

Faculté des sciences et Techniques
Département Mathématiques et Applications



Documentation portant sur le projet de collecte de livres
Programmation Orientée Objet

Responsable : M. CHARBONNEL Jacquelin

Binôme : ZAINABA Said Chaffi & AMOUZOU Efoe Prosper

I. Objectifs du projet

L'objectif de ce projet est de concevoir une application pour constituer et suivre une bibliothèque de livres. L'idée est de pouvoir collecter des livres sur le web (*web scraping*) pour constituer une bibliothèque, et générer divers catalogues de cette bibliothèque. On s'intéresse particulièrement aux livres au format EPUB et PDF. Mais l'application est bien extensible de façon à pouvoir facilement ajouter d'autres formats.

Projet collecte de livres, ce document détaille la structure et l'utilisation du projet de gestion de bibliothèque de livres. Le projet permet de gérer une collection de livres au format PDF et EPUB, d'en extraire des métadonnées, et de générer des rapports au format PDF. Cette documentation est destinée aux développeurs qui souhaitent utiliser ou étendre ce code.

II. Etape 1 du projet

Pour une bonne réalisation de cette première étape qui est l'étape fondamentale, nous avons procédé à l'implémentation des différentes classes principales :

➤ `base_livre`

Classe abstraite définissant l'interface commune pour les livres. Les méthodes obligatoires à implémenter sont : `'type'`, `'titre'`, `'auteur'`, `'langue'`, `'sujet'`, `'date'` qui retournent respectivement le type de livre, le titre du livre, l'auteur du livre, la langue du livre, le sujet du livre et la date de publication du livre.

➤ LivreEPUB

Hérite de `base_livre` représente un livre au format EPUB. Elle a comme attributs : ``ressource``, ``_titre``, ``_auteur``, ``_langue``, ``_sujet``, ``_date_publication``. Implémente les méthodes abstraites avec des données spécifiques.

➤ LivrePDF

Hérite de `base_livre`` également, représente un livre au format PDF. Et, elle a la même structure et de fonctionnalités similaires à LivreEPUB.

➤ base_bibli

Classe abstraite pour une bibliothèque de livres. Les méthodes à implémenter sont : ``ajouter``, ``rapport_livres``, ``rapport_auteurs`` qui respectivement ajoute un livre à la bibliothèque, génère un rapport listant les livres et un rapport listant les auteurs.

➤ simple_bibli

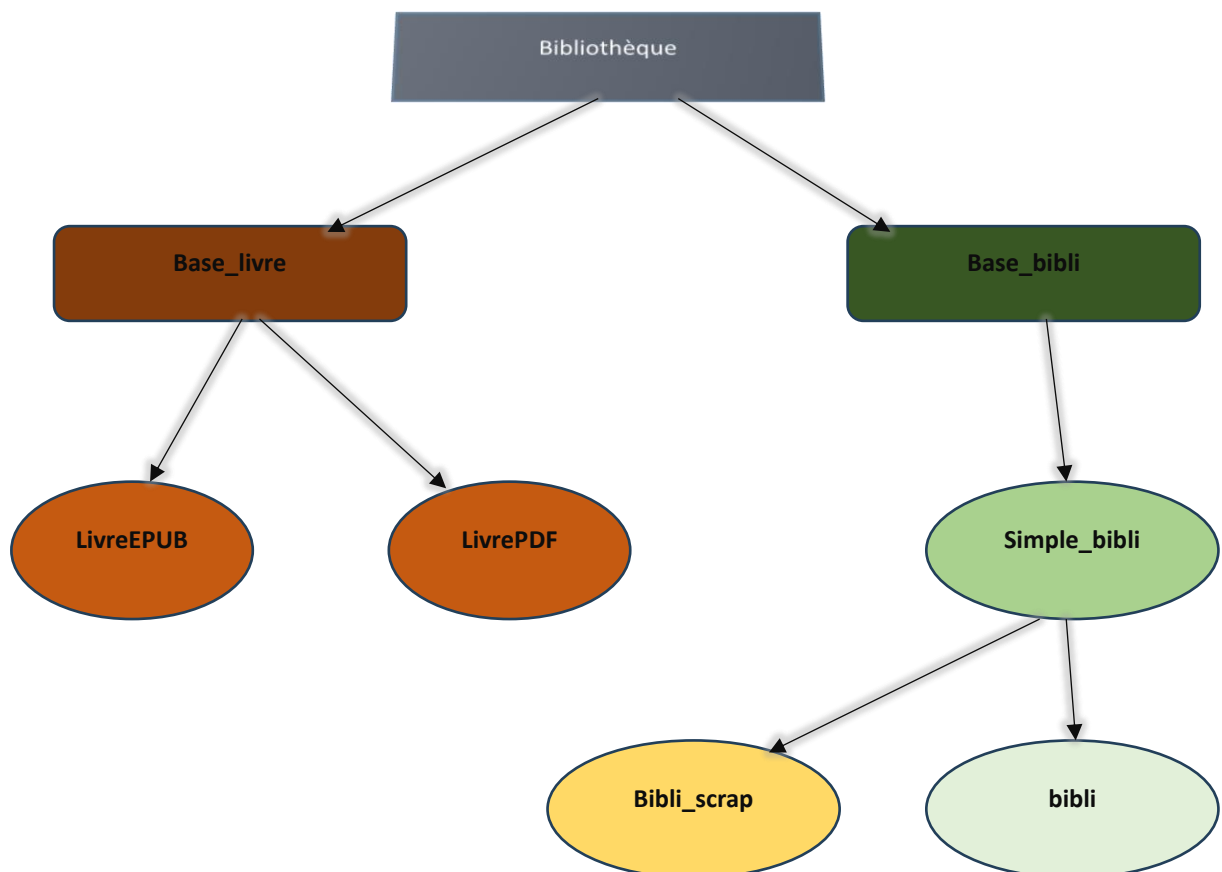
Hérite de `base_bibli`` et implémente une bibliothèque simple. Méthodes pour ajouter des livres et générer des rapports.

➤ bibli

Qui hérite de `simple_bibli``, permet d'alimenter la bibliothèque depuis des sources en ligne. Sa méthode principale est : ``alimenter`` qui télécharge et analyse les livres depuis une URL.

➤ bibli_scrap

Ajoute des fonctionnalités de scraping de documents sur le web. Les méthodes principales on a : ``scrap`` et ``download_documents``. `scrap(url, profondeur, nbmax)`: récupère des documents depuis le web selon la profondeur et le nombre maximum et `download_documents` télécharge les documents trouvés dans une page web.



Modules et Fonctions de Python

Plusieurs bibliothèques de python ont été importé et utilisées pour nos codes. Le tableau ci-dessous résume quelques-unes .

Bibliothèques Standard de python	os	datetime	requests	aiohttp	asyncio	BeautifulSoup (de bs4)
Détails	Permet la création des chemins d'accès pour stocker des fichiers téléchargés.	Permet la manipulation des dates et heures. Dans notre usage, elle assure la gestion des métadonnées telles que les dates de publication des livres.	Permet d'envoyer des requêtes HTTP (GET, POST, etc.) pour interagir avec des sites web ou des API	Télécharge simultanément plusieurs fichiers (PDF/EPUB) en utilisant des tâches asynchrones.	Coordination des fonctions asynchrones pour télécharger et traiter les fichiers.	Analyse et manipulation de documents HTML/XML . Ainsi, elle permet l'extraction des liens (PDF/EPUB) à partir des pages web

Exemples d'utilisation

- Initialisation d'une bibliothèque

```
bibliotheque = simple_bibli("chemin/vers/repertoire")
```

```
livre_pdf = LivrePDF("chemin/vers/livre.pdf", "Titre PDF", "Auteur", "Langue", "Sujet", "2024")
```

```
bibliotheque.ajouter(livre_pdf)
```

```
bibliotheque.rapport_livres("PDF", "rapport_livres.pdf")
```

- Téléchargement de livres depuis une URL

```
async def main():
```

```
    bibli = bibli("chemin/vers/repertoire")
```

```
    await bibli.alimenter("https://exemple.com/livres")
```

- Scraping de documents :

```
    scraper = scrap_bibli("chemin/vers/repertoire")
```

```
    await scraper.scrap("https://exemple.com", profondeur=2, nbmax=50).
```

Conclusion

Ce projet offre une base solide pour gérer des bibliothèques de livres électroniques. Les développeurs peuvent facilement étendre les classes pour ajouter de nouveaux formats ou des fonctionnalités avancées comme l'intégration à des bases de données ou des interfaces graphiques.