

# Documentation Technique

## Introduction

Cette documentation fournit des informations détaillées sur les classes Python créées pour gérer une bibliothèque de livres numériques. Ces classes sont destinées à être utilisées par des développeurs qui souhaitent créer ou étendre des fonctionnalités pour une application de gestion de bibliothèque.

## Classes Principales :

### ‘base\_livre’

La classe `base_livre` est une classe abstraite qui définit l'interface de base pour tous les types de livres. Les méthodes abstraites dans cette classe doivent être implémentées dans les sous-classes. Méthodes principales :

**type():** Renvoie le type du livre (EPUB, PDF, etc.).

**titre():** Renvoie le titre du livre.

**auteur():** Renvoie l'auteur du livre.

**Langue():** Renvoie la langue du livre.

**sujet():** Renvoie le sujet du livre.

**date():** Renvoie la date de publication du livre.

### ‘base\_bibli’

La classe `base_bibli` représente une bibliothèque de livres. Elle définit les méthodes que les sous-classes doivent implémenter pour ajouter des livres à la bibliothèque et générer des rapports sur les livres et les auteurs.

Méthodes principales :

**ajouter(livre):** Ajoute un livre à la bibliothèque.

**rapport\_livres(format, fichier):** Génère un rapport sur les livres de la bibliothèque.

**rapport\_auteurs(format, fichier):** Génère un rapport sur les auteurs des livres de la bibliothèque.

# Sous-Classes :

## ‘LivreEPUB’ et ‘LivrePDF’

Ces sous-classes de `base_livre` représentent respectivement les livres au format EPUB et PDF. Elles implémentent les méthodes abstraites de la classe parente.

## ‘simple\_bibli’

La classe `simple_bibli` est une implémentation simple de la bibliothèque. Elle étend `base_bibli` et stocke les livres dans une liste en mémoire. Les rapports générés contiennent des informations sur les livres et les auteurs au format spécifié.

**`ajouter(self, livre: base_livre)`** -> None: Ajoute le livre à la bibliothèque.

**`rapport_livres(self, format: str, fichier: str)`** -> None: Génère un rapport sur les livres de la bibliothèque et enregistre le résultat dans le fichier spécifié.

**`rapport_auteurs(self, format: str, fichier: str)`** -> None: Génère un rapport sur les auteurs des livres de la bibliothèque et enregistre le résultat dans le fichier spécifié.

## ‘Bibli’

La classe `Bibli` est une sous-classe de `simple_bibli` et est conçue pour gérer une bibliothèque de livres, permettant d'ajouter des livres à partir d'une URL spécifique. Pour instancier un objet `Bibli`, il faut fournir le chemin du fichier où la bibliothèque sera sauvegardée lors de la création de l'instance.

**`ajouter(self, livre)`** permet d'ajouter un objet de type `livre` à la liste de livres de la bibliothèque. Elle prend un argument, `livre`, qui doit être une instance de la classe `Livre` ou d'une de ses sous-classes.

**`alimenter(self, url)`** est utilisée pour récupérer des livres à partir d'une URL spécifiée. Elle effectue une requête HTTP sur l'URL fournie, analyse la page web pour extraire les informations pertinentes sur les livres disponibles, et ajoute ces livres à la bibliothèque en utilisant la méthode `ajouter`. Cette méthode utilise les bibliothèques externes `requests` pour effectuer la requête HTTP et `BeautifulSoup` pour analyser le contenu HTML de la page.

## ‘BibliScrap’

Cette classe permet de récupérer automatiquement des livres à partir d'une page web en suivant les liens HTML et répète le processus précédent sur chacune d'elles. Le processus se répète jusqu'à ce que l'un des critères d'arrêt de profondeur et `nbmax` soit vérifié.

**`scrap(self, url: str, profondeur: int, nbmax: int)`** -> None: Scrape la page web spécifiée pour récupérer des livres, en suivant les liens HTML jusqu'à la profondeur spécifiée ou jusqu'à ce que le nombre maximal de livres soit atteint. Utilise des méthodes auxiliaires pour télécharger les documents et vérifier les liens

**`HTML.is_html_link(self, a_tag: BeautifulSoup)`** -> bool: Vérifie si le lien pointe vers une page HTML. Utilisé pour filtrer les liens et ne récupérer que ceux qui pointent vers des pages

**HTML.download\_documents(self, soup: BeautifulSoup, url: str) -> None:** Télécharge les documents PDF et EPUB depuis la page web. Utilise la session de requête pour télécharger les fichiers et les stocke localement. Affiche également des messages de téléchargement réussi.