

Filière: Génie du Logiciel et des Systèmes Informatiques Distribués

Module : Ingénierie Multimédia - Développement Mobile

Année Universitaire : 2024/2025

TP : App de gestion de livres

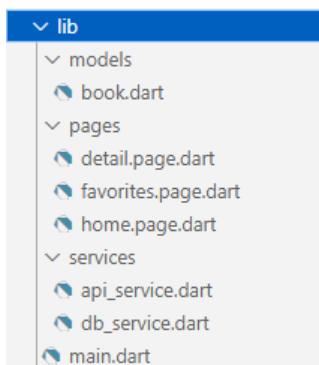
(API + SQLite)

Objectif :

Créer une application Flutter qui permet :

1. De **chercher des livres** via une API publique (ex : Google Books API)
2. D'**enregistrer des livres favoris** localement (titre, auteur, image)
3. De **consulter et gérer la liste des favoris** (avec possibilité de suppression)

Structure du projet et Fonctionnalités à implémenter :



	Fonctionnalité	API / SQLite
	Rechercher un livre par mot-clé	API
	Afficher les résultats (titre, auteur, image)	API
	Ajouter un livre à la liste des favoris	SQLite
	Afficher la liste des livres favoris	SQLite
	Supprimer un livre des favoris	SQLite

API suggérée :

- **Google Books API** (exemple d'appel) :
<https://www.googleapis.com/books/v1/volumes?q=flutter>

(N.B.: q= est le mot clé à chercher « query »)

Packages nécessaires :

Exécuter cette commande pour ajouter les package dans le fichier « *pubspec.yaml* »

```
>> flutter pub add http sqflite path
```

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  http: ^1.4.0  
  sqflite: ^2.4.2  
  path: ^1.9.1
```

Instructions à suivre:

- Modèle Book.dart :

```
● ● ●  
  
class Book {  
  final String id; // Peut être un ISBN ou autre identifiant unique  
  final String title;  
  final String author;  
  final String imageUrl;  
  
  Book(...);  
  
  // Convertir un JSON (API) vers un objet Book  
  factory Book.fromJson(Map<String, dynamic> json) {  
    return Book(  
      ...  
    );  
  }  
  
  // Pour stocker dans SQLite (Map)  
  Map<String, dynamic> toMap() {  
    return {  
      ...  
    };  
  }  
  
  // Pour créer un Book à partir d'un Map SQLite  
  factory Book.fromMap(Map<String, dynamic> map) {  
    return Book(  
      ...  
    );  
  }  
}
```

- **Implémenter les fonctions dans db_service.dart**

- insertItem(item) : pour ajouter un favori
- getItems() : pour lire les favoris
- deleteItem(id) : pour supprimer un favori

- **Interface utilisateur:**

- Utilisez FutureBuilder pour récupérer et afficher les données en asynchrone.
- Créez une nouvelle page FavoritesPage avec une ListView/Gridview pour :
 - Afficher les éléments sauvegardés
 - Supprimer un élément avec un bouton 



```
GridView.builder(
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 2),
    itemCount: items.length,
    itemBuilder: (context, index) {
        return Card(
            child: Column(
                children: [
                    Image.network(items[index].imageUrl), // chargement d'image
                    Text(items[index].title),
                    IconButton(
                        icon: Icon(Icons.favorite_border),
                        onPressed: () {
                            // Enregistrer dans SQLite
                        },
                    ),
                    ],
                ),
            );
    },
);
```

- **Bouton d'ajout:** Sur chaque livre, ajoutez un bouton  :
 - Si l'élément est déjà sauvegardé →  plein
 - Sinon →  vide
 - Action : insérer ou supprimer l'élément de la base locale.