

# Rapport de Projet - API de Gestion de Livres

## 1. Introduction

Dans le cadre de ce projet, nous avons développé une API permettant de gérer une base de données de livres. Cette API offre des fonctionnalités CRUD (Create, Read, Update, Delete) pour interagir avec les informations relatives aux livres, telles que le titre, l'auteur, la date de publication, etc. L'objectif principal de cette API est de fournir une interface standardisée et sécurisée pour les applications clientes (sites web, applications mobiles, etc.) afin de manipuler les données des livres de manière efficace et fiable.

## 2. Stack Technologique

Pour le développement de cette API, nous avons choisi d'utiliser les technologies suivantes :

- **Langage** : PHP, HTML, CSS
- **Framework** : Electron (de Visual Studio Code)
- **Base de données** : MySQL (via phpMyAdmin)

Le choix du langage PHP nous a permis de bénéficier de sa popularité, de sa maturité et de sa vaste communauté de développeurs. L'utilisation de HTML et CSS nous a permis de créer une interface utilisateur attrayante et responsive. Quant au framework, nous avons sélectionné Electron, qui nous a permis de développer une application de bureau multiplateformes à partir de technologies web (HTML, CSS, JavaScript). Enfin, la base de données MySQL a été retenue pour sa fiabilité, sa facilité d'utilisation et son excellente intégration avec PHP.

## 3. Conception de l'API

L'API de gestion de livres comprend les endpoints suivants :

### 3.1. Création d'un livre

**Méthode** : POST

**Endpoint** : `http://localhost/API-BOOK/api/create.php`

**Réponse** : Un message indiquant que le livre a été ajouté avec succès, avec son nouvel ID.

### 3.2. Récupération de la liste des livres

**Méthode** : GET

**Endpoint** : `http://localhost/API-BOOK/api/read.php`

**Réponse** : Un tableau contenant les informations de chaque livre, affiché sous forme de table HTML.

### 3.3. Mise à jour d'un livre

**Méthode** : POST

**Endpoint** : `http://localhost/API-BOOK/api/update.php`

**Réponse :** Un message indiquant que le livre a été mis à jour avec succès.

### 3.4. Suppression d'un livre

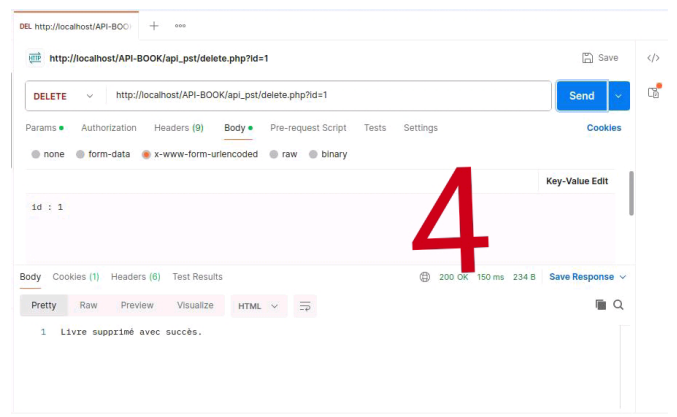
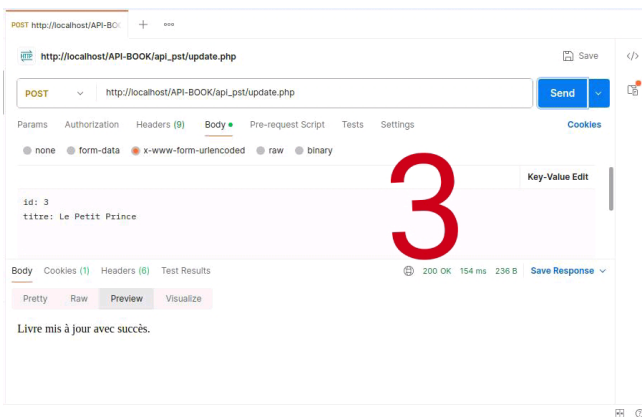
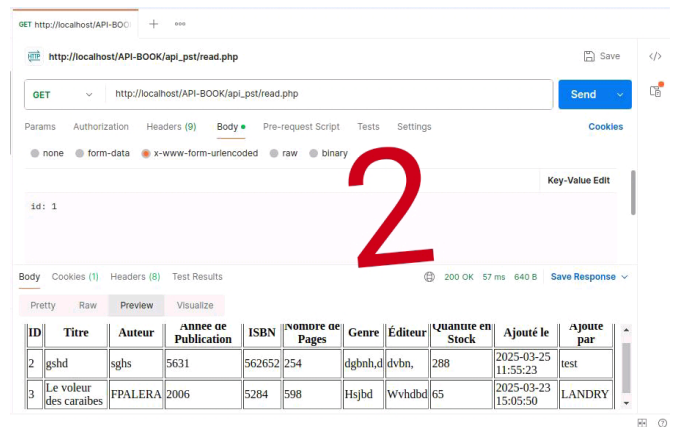
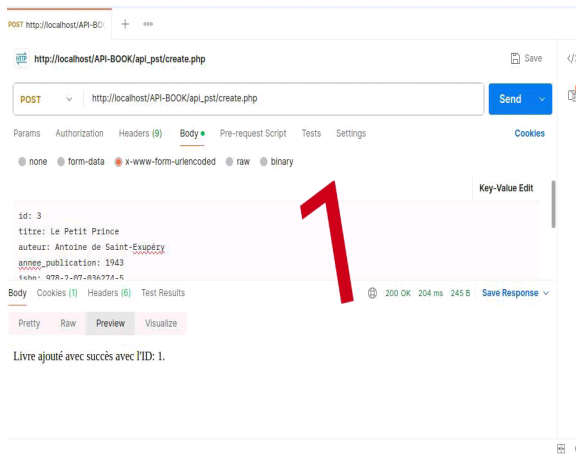
**Méthode :** DELETE

**Endpoint :** `http://localhost/API-BOOK/api/delete.php`

**Réponse :** Un message indiquant que le livre a été supprimé avec succès.

## 4. Tests avec Postman

Détails des tests effectués ici.



## 5. Difficultés Rencontrées et Solutions

- **Gestion de la date de publication** : Initialement, nous avons choisi de stocker la date de publication sous forme de chaîne de caractères dans la base de données...
- **Sécurité et authentification** : Nous avons ajouté une couche de sécurité...
- **Requêtes Postman** : La méthode d'authentification empêchait les requêtes de s'exécuter...
- **Gestion des erreurs** : Nous avons dû mettre en place une gestion robuste des erreurs...

## 6. Conclusion

Ce projet de développement d'une API de gestion de livres a été une expérience enrichissante. Nous avons pu mettre en pratique nos compétences en programmation PHP, en conception d'interfaces web avec HTML et CSS, ainsi que notre capacité à utiliser MySQL via phpMyAdmin et à développer une application de bureau avec le framework Electron.

**Répertoire Github** : <https://github.com/FPALERA/API-BOOK>