

TP 2 : Création d'une API avec [Express.js](#) et MySQL

Durée estimée : 3 heures

Créer une API complète pour gérer une bibliothèque de livres en utilisant Express.js et MySQL, en implémentant toutes les opérations CRUD.

1 - Configuration du projet et base de données

Créer un nouveau projet Node.js avec Express.

Configurer la connexion à MySQL

Créer une base de données library avec une table books contenant les champs :

- id (INT, AUTO_INCREMENT, PRIMARY KEY)
- title (VARCHAR)
- author (VARCHAR)
- isbn (VARCHAR)
- publication_year (INT)
- created_at (TIMESTAMP)

2 - Routes CRUD de base

Implémenter les routes suivantes pour la gestion des livres :

GET /api/books

- Récupérer tous les livres
- Ajouter la pagination (limite et offset)
- Ajouter le tri par titre ou année de publication

GET /api/books/:id

- Récupérer un livre spécifique
- Gérer le cas où le livre n'existe pas

POST /api/books

- Créer un nouveau livre
- Valider les données reçues
- Retourner le livre créé avec son ID

PUT /api/books/:id

- Mettre à jour un livre existant
- Valider les données reçues
- Gérer le cas où le livre n'existe pas

DELETE /api/books/:id

- Supprimer un livre
- Gérer le cas où le livre n'existe pas

3 - Fonctionnalités avancées

Ajouter la recherche :

- Créer une route GET /api/books/search
- Permettre la recherche par titre, auteur ou ISBN
- Implémenter la recherche partielle (LIKE)

Ajouter les statistiques

- Créer une route GET /api/books/stats
- Afficher le nombre total de livres
- Afficher le nombre de livres par année de publication
- Afficher le nombre de livres par auteur