



ANALYSE DE DONNÉES D'UN SITE DE LIVRES

bookstoscrap

MOHAMED, MATHIEU

Content

01

Introduction

02

Scraping des Données

03

Analyse des Données

04

Visualisation des Données

05

Interprétation des Résultats

06

Déploiement

07



introduction

OBJECTIFS DU PROJET:

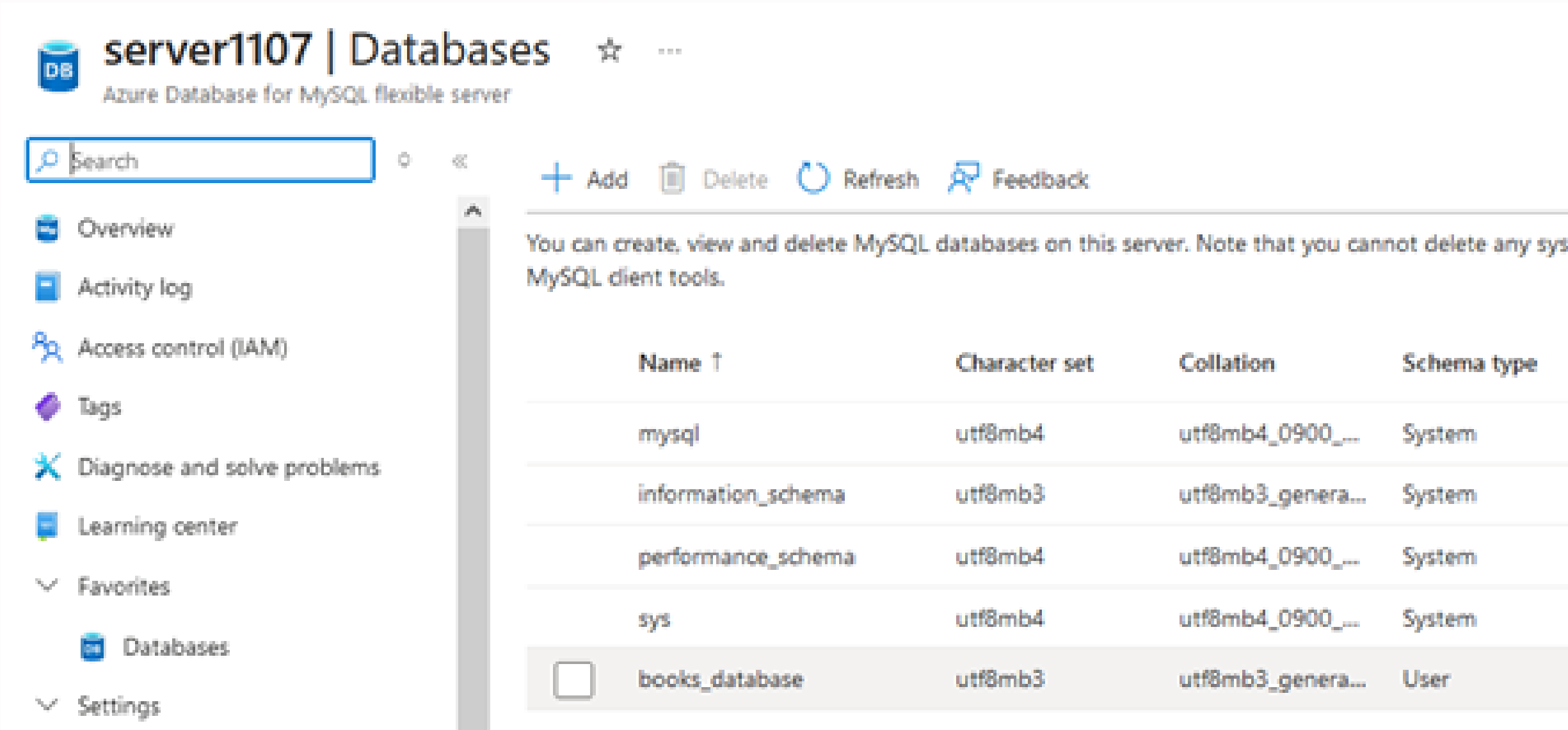
- COLLECTER DES DONNÉES DE LIVRES EN LIGNE
- ANALYSER LES DONNÉES POUR OBTENIR DES INSIGHTS
- STOCKER LES DONNÉES DANS UNE BASE DE DONNÉES MYSQL
- DÉPLOYER LE PROJET SUR AZURE POUR L'ACCESSIBILITÉ

scrapping

- **TECHNIQUES DE SCRAPING UTILISÉES**
- **REQUESTS.GET() POUR TÉLÉCHARGER LES PAGES WEB**
- **BEAUTIFULSOUP POUR PARSE LE CONTENU HTML**
- **EXEMPLES DE CODE DE SCRAPING**
- **DONNÉES EXTRAITES (TITRE, CATÉGORIE, DISPONIBILITÉ, PRIX, ETC.)**

Création serveur MySQL

- Créer nouvelle ressource sur le Azure Dashboard
- Sélectionner Serveur MySQL
- Autoriser les connexions extérieures
- Ajouter l'IP de travail à la liste blanche



The screenshot displays the Azure portal interface for a MySQL flexible server named 'server1107'. The left sidebar contains navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Learning center, Favorites, Databases, and Settings. The main content area shows a search bar and action buttons: Add, Delete, Refresh, and Feedback. Below these, a message states: 'You can create, view and delete MySQL databases on this server. Note that you cannot delete any system MySQL client tools.' A table lists the existing databases:

Name ↑	Character set	Collation	Schema type
mysql	utf8mb4	utf8mb4_0900_...	System
information_schema	utf8mb3	utf8mb3_genera...	System
performance_schema	utf8mb4	utf8mb4_0900_...	System
sys	utf8mb4	utf8mb4_0900_...	System
<input type="checkbox"/> books_database	utf8mb3	utf8mb3_genera...	User

Création base de données MySQL



server1107 | Databases



Azure Database for MySQL flexible server



Add



Delete



Refresh



Feedback



Overview



Activity log



Access control (IAM)



Tags



Diagnose and solve problems



Learning center



Favorites



Databases

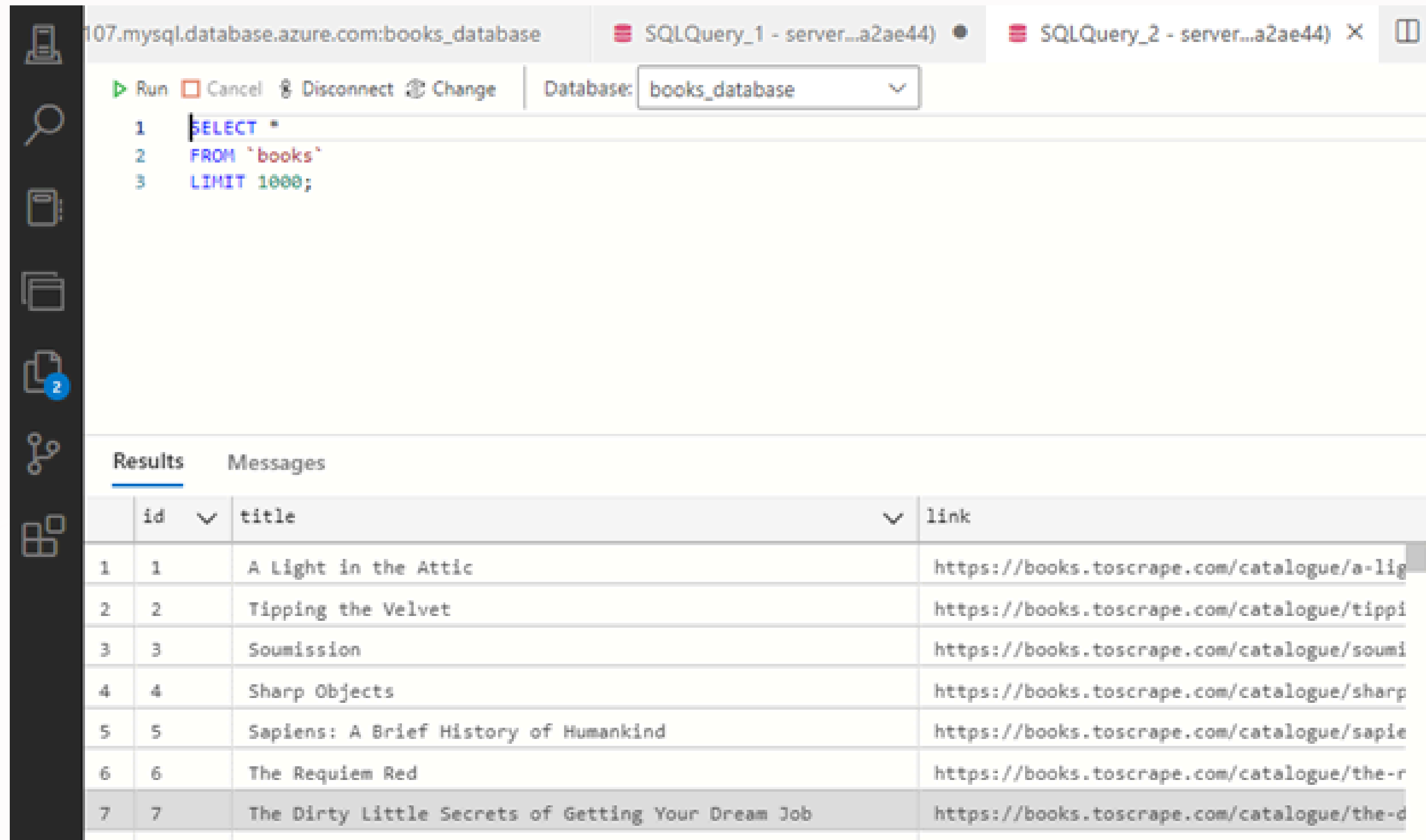


Settings

You can create, view and delete MySQL databases on this server. Note that you cannot delete any system schemas such as mysql.sys, information_schema, performance_schema, sys, and mysql.sys, in MySQL client tools.

Name	Character set	Collation	Schema type	
mysql	utf8mb4	utf8mb4_0900_...	System	
information_schema	utf8mb3	utf8mb3_genera...	System	
performance_schema	utf8mb4	utf8mb4_0900_...	System	
sys	utf8mb4	utf8mb4_0900_...	System	
<input type="checkbox"/>	books_database	utf8mb3	utf8mb3_genera...	User Open in Power BI

Explorer les données



The screenshot shows a web-based SQL client interface. At the top, there are tabs for the database connection: '107.mysql.database.azure.com:books_database', 'SQLQuery_1 - server...a2ae44)', and 'SQLQuery_2 - server...a2ae44)'. Below the tabs, there is a toolbar with buttons for 'Run', 'Cancel', 'Disconnect', and 'Change'. A dropdown menu shows the selected database as 'books_database'. The main area contains a SQL query:

```
1 SELECT *
2 FROM `books`
3 LIMIT 1000;
```

Below the query editor, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

	id	title	link
1	1	A Light in the Attic	https://books.toscrape.com/catalogue/a-lig
2	2	Tipping the Velvet	https://books.toscrape.com/catalogue/tippi
3	3	Soumission	https://books.toscrape.com/catalogue/soumi
4	4	Sharp Objects	https://books.toscrape.com/catalogue/sharp
5	5	Sapiens: A Brief History of Humankind	https://books.toscrape.com/catalogue/sapie
6	6	The Requiem Red	https://books.toscrape.com/catalogue/the-r
7	7	The Dirty Little Secrets of Getting Your Dream Job	https://books.toscrape.com/catalogue/the-d

Création API

```
def get_books():
    category = request.args.get('category')
    limit = request.args.get('limit', default=10, type=int)

    print(f"Received request for genre: {category} with limit: {limit}")

    # Requête pour obtenir les livres
    books = Book.query.filter_by(category=category).limit(limit).all()

    # Convertir les résultats en JSON
    books_list = []
    for book in books:
        books_list.append({
            'title': book.title,
            'link': book.link,
            'price': book.price,
            'availability': book.availability,
            'stars': book.stars,
            'category': book.category
        })

    return jsonify(books_list)
```


Utilisation API

- Test en local de l'API avec PostMan
- Plus sécurisé que les requêtes SQL

The screenshot shows the Postman interface for a POST request to `http://127.0.0.1:5000/books?category=fiction&limit=10`. The request is saved and ready to be sent. The query parameters are defined as follows:

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	category	fiction	
<input checked="" type="checkbox"/>	limit	10	
	Key	Value	

The response status is 200 OK, with a response time of 417 ms and a size of 2.47 KB. The response body is displayed in the 'Pretty' view as a JSON array:

```
{
  "category": "Fiction",
  "link": "https://books.toscrape.com/catalogue/private-paris-private-10_958/index.html",
  "price": 47.61,
  "stars": 5,
  "title": "Private Paris (Private #10)"
},
{
  "availability": "In stock",
  "category": "Fiction",
  "link": "https://books.toscrape.com/catalogue/"
```