Documenatación CRUD en Flask

Tabla de contenido

[1. Requerimientos 2](#_Toc169200264)

[2. Estructura de archivos y carpetas 2](#_Toc169200265)

[3. APIS POSTMAN 2](#_Toc169200266)

[3.1. /books (GET): Devuelve una lista de todos los libros disponibles. 2](#_Toc169200267)

[3.2. /books/<int:book\_id> (GET): Devuelve los detalles de un libro específico basado en su ID. 2](#_Toc169200268)

[3.3. /books (POST): Crea un nuevo libro en la colección. 2](#_Toc169200269)

[3.4. /books/<int:book\_id> (PUT): Actualiza los detalles de un libro existente. 2](#_Toc169200270)

[3.5. /books/<int:book\_id> (DELETE): Elimina un libro de la colección. 3](#_Toc169200271)

# Requerimientos

* Python 3.9.13
* Pip 22.3.1
* Ambiente virtual
  + Para esto solo es correr el comando: python -m venv venv
  + Se ingresa al ambiente con el comando: .\venv\Scripts\activate
* Flask
  + Se instala con el comando: pip install flask
* Comandos necesarios:
  + Arranque de aplicación: Python src/main.py
* Flask-pymongo
  + Paquete para realiza conexión y petición a base de datos mongo: pip install flask-pymongo python-dotenv
* Dotenv
  + Paquete para utilizar variables de entorno: pip install Python-dotenv

# Estructura de archivos y carpetas

* Src

conexión a la base de datos

* + databa
    - db

**Configuración de rutas**

* routes
  + booksRoutes

**Servicios transaccionales con las distintas peticiones**

* services
  + booksServices

**Simplemente un archivo HTML que responde a la ruta base del api**

* templates
  + index.html

**Archivo Bootstrap que inicializa nuestro proyecto**

* main.py

**Archivo para administrar Environment Variables**

* **.env**

# APIS POSTMAN

## /books (GET): Devuelve una lista de todos los libros disponibles.

<http://127.0.0.1:5000/books>

## /books/<int:book\_id> (GET): Devuelve los detalles de un libro específico basado en su ID.

<http://127.0.0.1:5000/books/666b8ebb309d6042060a4a10>

## /books (POST): Crea un nuevo libro en la colección.

<http://127.0.0.1:5000/books>

{

"author": "me",

"title": "the mouse",

"description": "book with pages"

}

## /books/<int:book\_id> (PUT): Actualiza los detalles de un libro existente.

<http://127.0.0.1:5000/books/666b8e11a35da1c60c865b5c>

{

"author": "me",

"title": "the kayboard",

"description": "book with keys"

}

## /books/<int:book\_id> (DELETE): Elimina un libro de la colección.

<http://127.0.0.1:5000/books/666b8e11a35da1c60c865b5c>