

Book Service API

Introduction

Welcome to the Book Service API guide! This easy-to-follow tutorial will help you create and run a FastAPI project on your Windows PC. With this API, you can efficiently manage books, authors, and clients for your book-related activities.

Capabilities

- Add a book.
- Edit book details.
- Filter books by title and author.
- Add multiple books by the same author.
- Add an author.
- Create a client.
- Track books borrowed by a client.
- Link and unlink a client to/from a book.

Installation on Windows

1. Set Up the Project Directory

Open the Command Prompt or PowerShell and create a new folder for your project:

bash

mkdir book-service

cd book-service

2. Install Dependencies

Create a virtual environment and install the necessary packages:

bash

```
python -m venv venv  
.\venv\Scripts\activate  
pip install fastapi[all] psycpg2 sqlalchemy python-decouple
```

3. Configuration File

Create a `.env` file in the project directory and add the following:

env

```
DATABASE_URL=postgresql://book_user:book_password@localhost/book_db  
SECRET_KEY=your-secret-key
```

Setting Up FastAPI on Windows

1. Create the FastAPI Application

Create a `main.py` file in the project directory and copy the provided code into it.

#Docker Compose Configuration on Windows

1. Configure PostgreSQL

Create a `docker-compose.yml` file in the project directory with the following:

yaml

version: '3.8'

services:

postgres:

image: postgres:latest

environment:

POSTGRES_DB: book_db

POSTGRES_USER: book_user

POSTGRES_PASSWORD: book_password

ports:

- "5432:5432"

2. Configure FastAPI

Extend the `docker-compose.yml` file to include FastAPI:

yaml

version: '3.8'

services:

postgres:

image: postgres:latest

environment:

POSTGRES_DB: book_db

POSTGRES_USER: book_user

POSTGRES_PASSWORD: book_password

ports:

- "5432:5432"

fastapi:
build: .
command: uvicorn main:app --reload --host 0.0.0.0 --port 8000
volumes:
- .\app:/app
ports:
- "8000:8000"
depends_on:
- postgres

Implementing FastAPI Endpoints on Windows

1. Adding a Book

To add a book, use the `/books/` endpoint. Send a POST request with the book details.

2. Editing a Book

To edit a book, use the `/books/{book_id}` endpoint. Send a PUT request with the `book_id` and updated details.

#Running the Services on Windows

1. Start PostgreSQL

Run the following command in the project directory:

bash

docker-compose up --build

2. Start FastAPI

Visit <http://localhost:8000/docs> to access the FastAPI documentation and test your API.

#Testing the API on Windows

1. Accessing FastAPI Documentation

Visit <http://localhost:8000/docs> to access the FastAPI documentation.

2. Testing Endpoints

Use the provided Swagger UI or tools like `curl` or [Postman](<https://www.postman.com/>) to test the API endpoints.

Additional Considerations

1. Security

- The API is secured using Bearer authentication. The token list is static and can be modified in the code.

2. Production Deployment

- For a production environment, deploy the application with proper security measures, including HTTPS and a reverse proxy.