Address Book Django App – Documentation

Project Overview

This is a **Django-based Address Book application** with the following features:

- **JWT Authentication:** Secure login and registration for users.
- **Address Management:** Users can add and view addresses with latitude and longitude.
- **Pagination:** Efficiently browse large lists of addresses.

The app is designed to be easy to run locally and extendable for future development.

Prerequisites

Make sure you have the following installed on your system:

- Python 3.8 or higher
- pip (Python package manager)
- Git

Step 1: Clone the Repository

Open your terminal and run:

```
git clone https://github.com/deepa353820/Address_book.git
cd Address_book
```

Step 2: Create a Virtual Environment (Recommended)

```
python -m venv venv
```

Activate the virtual environment:

- **Windows:**

```
venv\Scripts\activate
```

- **Mac/Linux:**
- `source venv/bin/activate`

Step 3: Install Dependencies

```
pip install -r requirements.txt
```

This installs all necessary Python packages for the app.

Step 4: Apply Database Migrations

```
python manage.py migrate
```

This sets up the database tables required by Django.

Step 5: Create a Superuser (Optional)

```
python manage.py createsuperuser
```

Follow the prompts to set up an admin account.

Step 6: Run the Development Server

```
python manage.py runserver
```

By default, the app will be available at:
`http://127.0.0.1:8000/`

Step 7: Using the API Endpoints

The app provides REST API endpoints (secured with JWT) for:

- **User Authentication:** `/api/token/` (login), `/api/token/refresh/`
- **Address Management:** `/api/addresses/` (add/view addresses)

Pagination is enabled by default for listing addresses.

Step 8: Access Django Admin (Optional)

If you created a superuser:
`http://127.0.0.1:8000/admin`

Use your admin credentials to log in.

API Documentation

The project includes **Swagger UI documentation** for all API endpoints. Once the server is running locally, you can explore and test the API interactively at:

```
http://127.0.0.1:8000/swagger/
```

This Swagger UI provides detailed information about each endpoint, including:

- Available HTTP methods (GET, POST, PUT, PATCH, DELETE)
- Required parameters and request bodies
- Authentication requirements
- Response formats

It's a helpful tool for quickly understanding and testing the API without external tools.

Notes

- Ensure `.env` or environment variables are set if your project uses secrets (optional).
- JWT authentication requires tokens for accessing protected endpoints.