# Django REST Framework Datastore Backend API Readme

A brief description of what this project does and who it's for.

## Table of Contents

## Overview

This Django REST Framework Datastore Backend API provides a set of endpoints that allow you to interact with a Datastore backend. It enables you to create, read, update, and delete data from the Datastore. The API is designed to be lightweight and easy to use, allowing developers to quickly get started with their projects.

## Table of Contents

- [Features](#features)

- [Prerequisites](#prerequisites)

- [Installation](#installation)

- [Usage](#usage)

- [Examples](#examples)

- [License](#license)

## Features

- CRUD operations as specified by the brief (Create, Read, Delete) for Datastore entities

- Pagination and filtering for large result sets

- Error handling and response codes

- JSON response formats

- Secure authentication and authorization

## Prerequisites

- Python (>= 3.11.x)

## Installation

1. Clone the repository from Git: `git clone https://github.com/JDGresse/1\_Dev\_Hut\_Forum.git`

2. Create and activate a virtual environment: `python3 -m venv env` and `source env/bin/activate`

3. Install the required packages: `pip install -r requirements.txt`

4. Run the server: `python manage.py runserver`

5. This creates a local server at <http://127.0.0.1:8000/>

## Usage

To use the Forum Django REST Framework Datastore Backend API, send requests to the following endpoints:

- `/admin/`: Open admin page

- `/posts/`: Get a list of posts

- `/posts/{id}/`: Get a post by ID

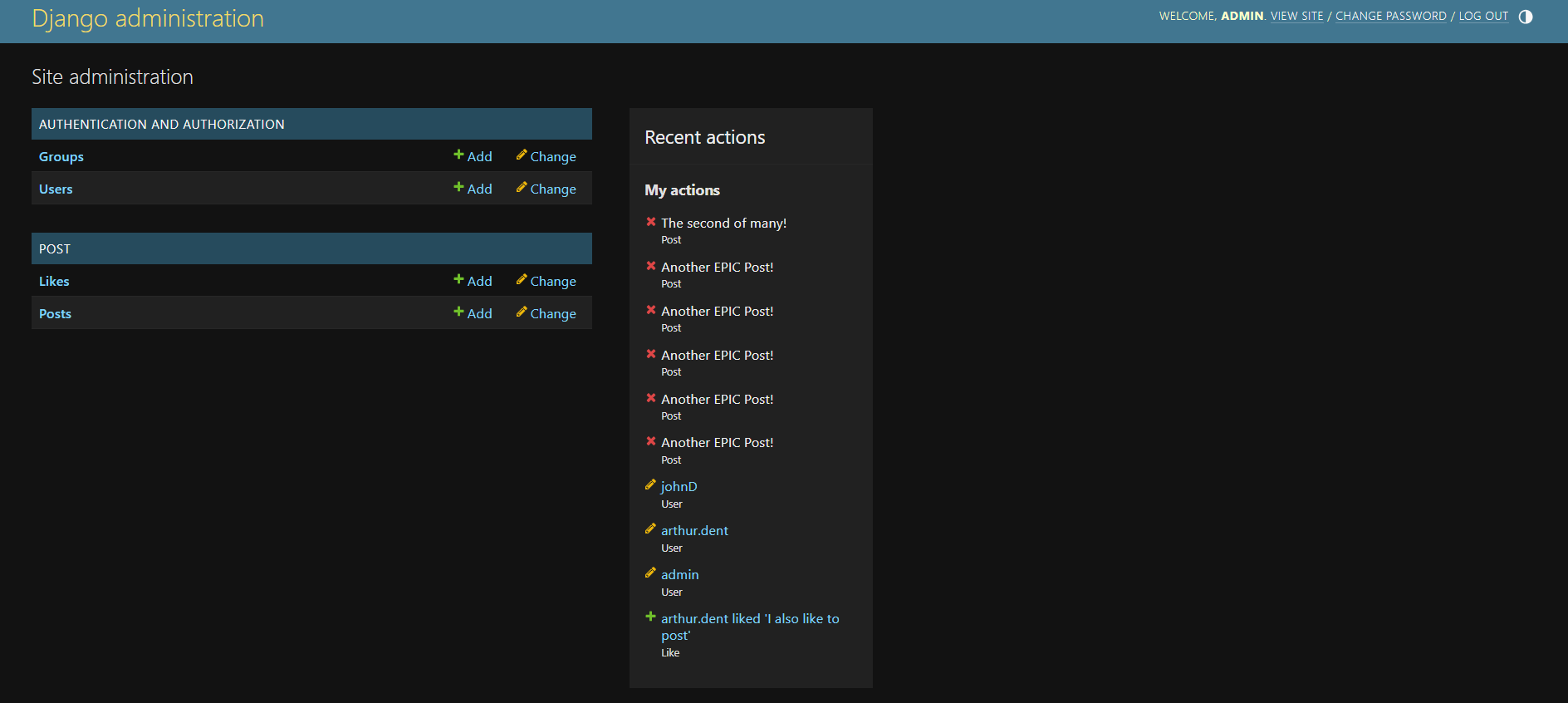
- `/posts/likes/`: Get a list of likes

## Examples

### - `/admin/`: Open admin page

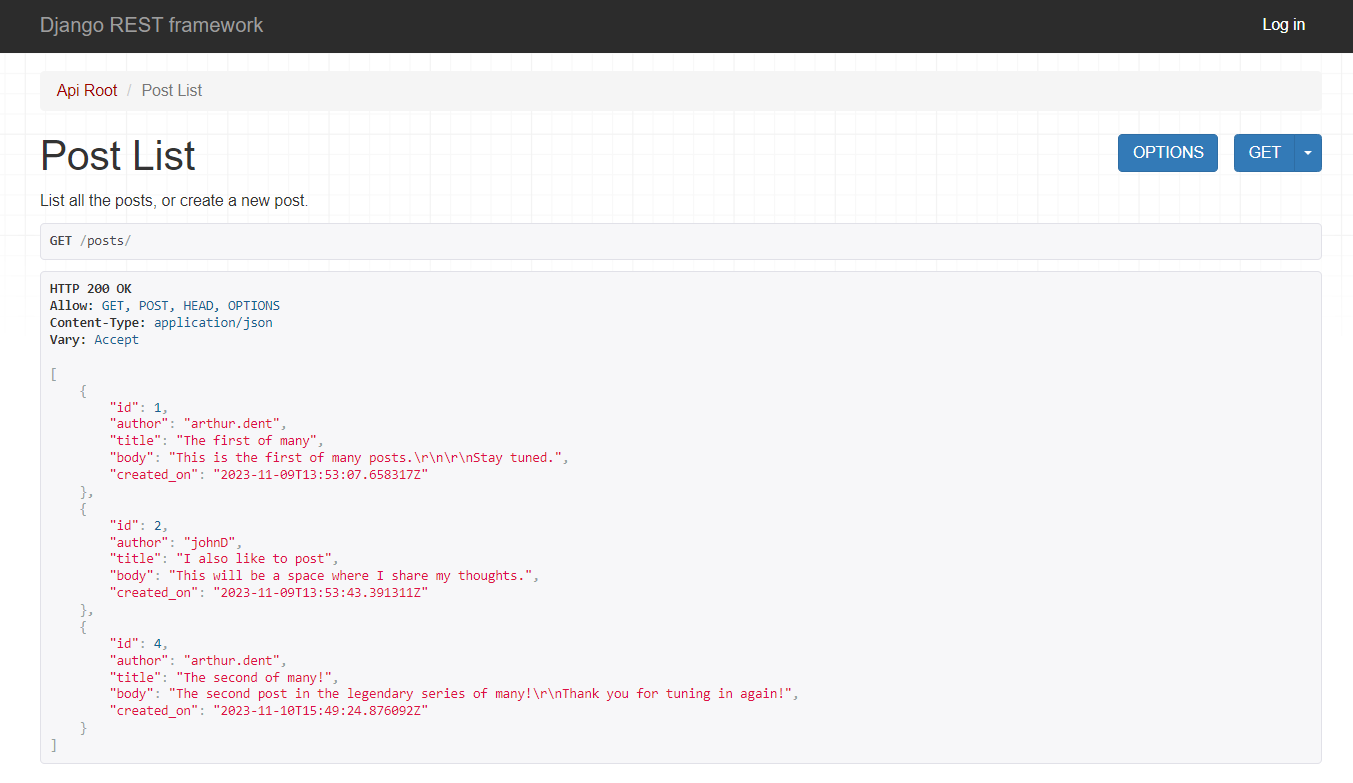
A superuser – admin – is created (login details to be requested from repository owner).

The Django administration page - `/admin/`:



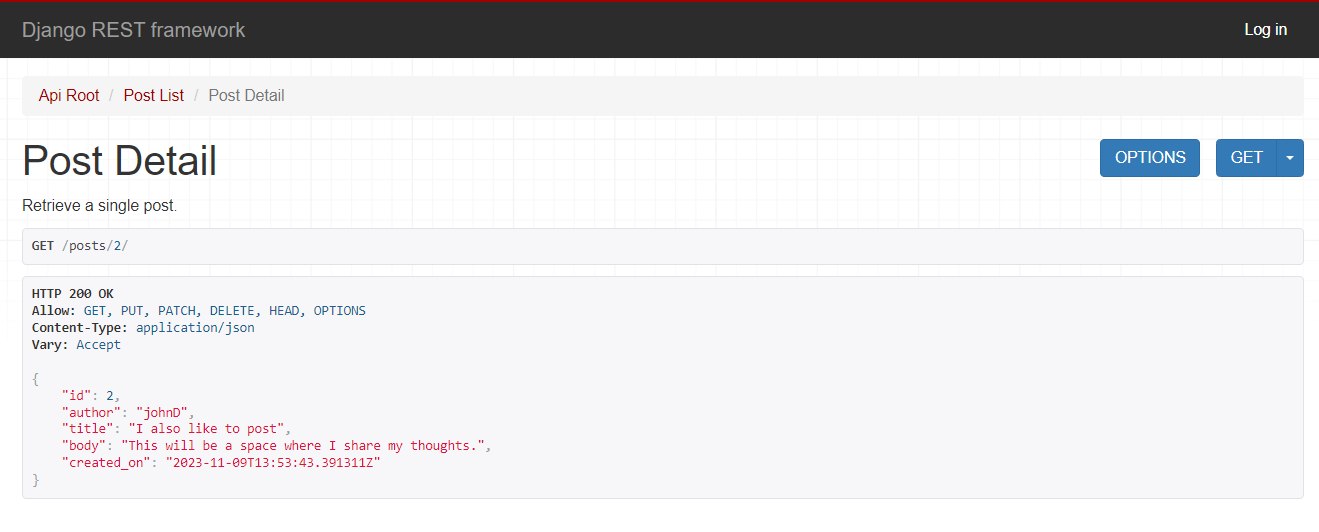
### - `/posts/`: Get a list of posts

An example of the read-only post list page. The list is read-only for anyone viewing the URL, with only authenticated users able to create new posts.



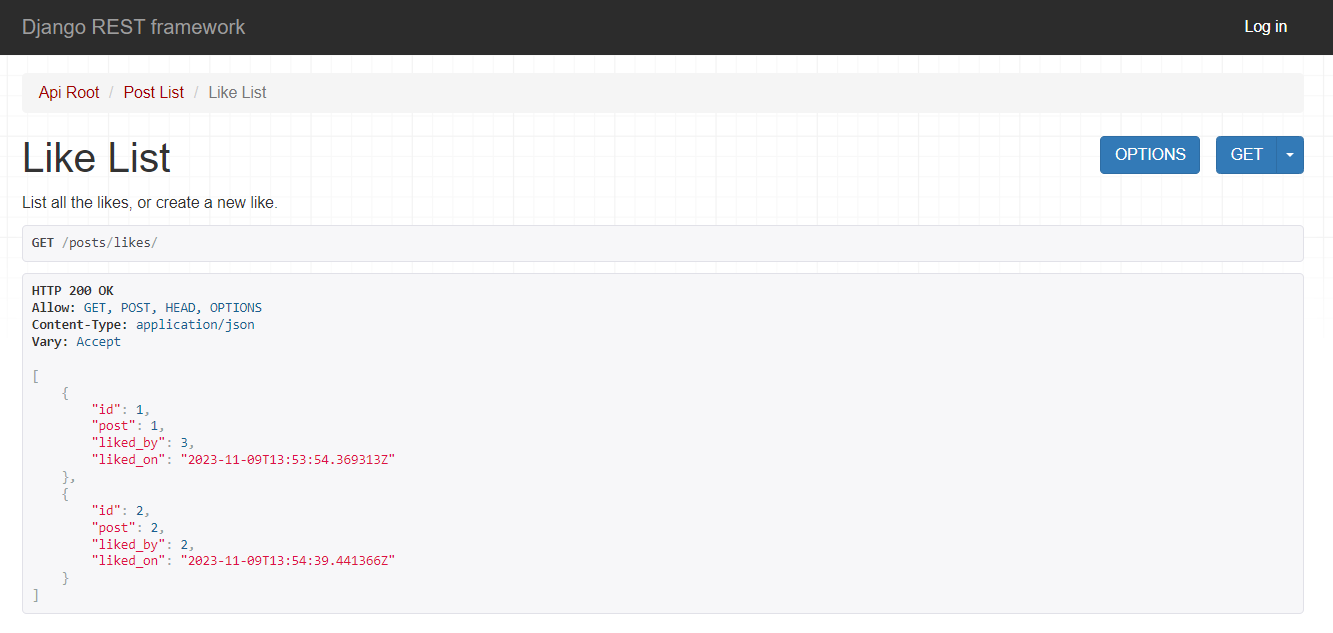
Users can view specific posts with

### - `/posts/{id}/`: Get a post by ID



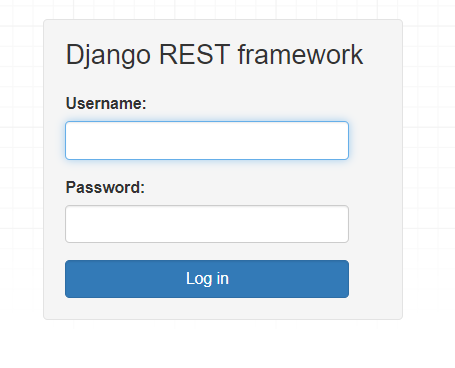
### - `/posts/likes/`: Get a list of likes

An example of the read-only likes list page. The list is read-only for anyone viewing the URL, with only authenticated users able to create new likes.



### User login:

The top right corner of the page contains a ‘login’ link for authorised users:



Currently includes two authorized users, with details available from the repository owner.

## Testing

1. Install Postman: [Download Postman](https://www.postman.com/downloads/)

2. Import the Postman collection: `Forum-Test-Collection.postman\_collection.json`

3. Run the collection using the Collection Runner in Postman.

4. Check the test results in the Postman console.

## License

* TBD.