151104

Wang'ondu Claire Wambui

Django Blogsite API – Assignment Report

1. Overview

In this assignment, I extended the existing **Django Blogsite API** by implementing CRUD functionality for:

- **Posts**
- **Comments**
- **Categories**
- **Profiles**

The API was built using **Django REST Framework** and included authentication and permission controls.

2. Features Implemented

Models

- **Post** – Blog posts created by users.
- **Comment** – Comments linked to blog posts.
- **Category** – Categories to organize blog posts.
- **Profile** – Extended user profile model.

Serializers

Created serializers for all models:

- PostSerializer
- CommentSerializer
- CategorySerializer
- ProfileSerializer

ViewSets

- PostViewSet → Requires IsAuthenticated
- CommentViewSet → Requires IsAuthenticated
- CategoryViewSet → Requires IsAdminUser
- ProfileViewSet → Requires IsAuthenticated

Routing

Configured DefaultRouter in **urls.py** to automatically generate routes:

bash

Copy code

```
/api/posts/

/api/comments/

/api/categories/

/api/profiles/
```

3. Permissions

- Only authenticated users can create and view **posts**, **comments**, and **profiles**.
- Only admins can create, update, or delete **categories**.

4. Testing Process

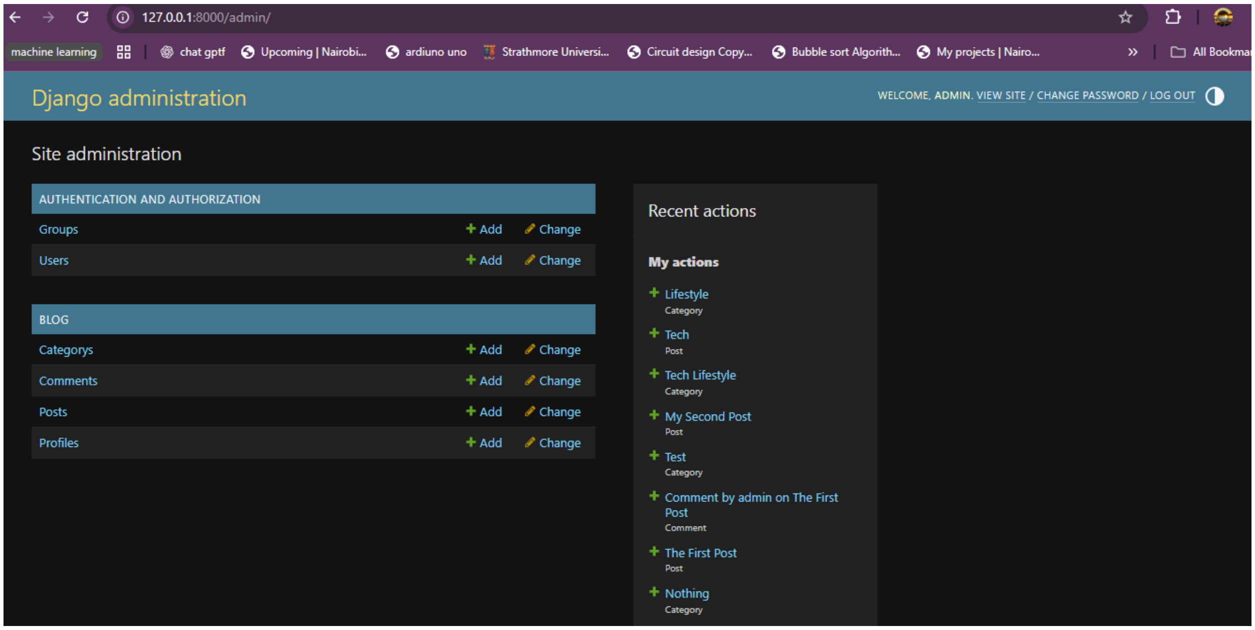
Authentication

1. Created users in the Django shell:
    - alice (normal user)
    - editor (normal user)
    - staff (staff member)
    - moderator (normal user)
2. Logged in via **Django Admin** to confirm authentication.

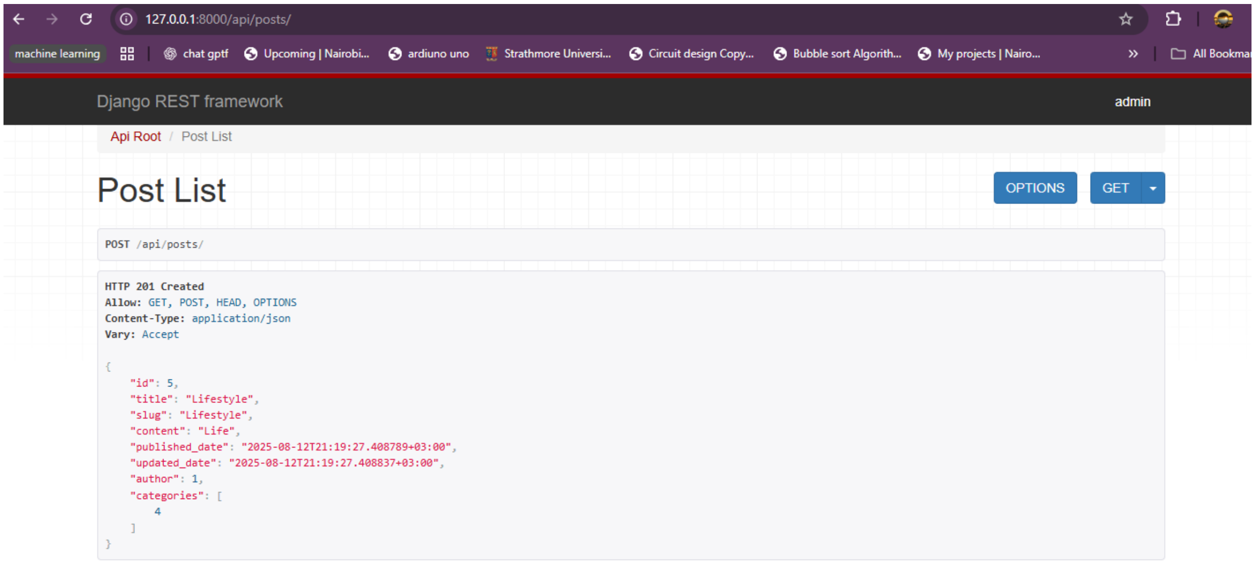
Endpoints Tested

- **GET** requests for all models (successful with authentication).
- **POST** requests to create new entries (successful with authentication).
- Accessing /api/categories/ without admin rights returned **403 Forbidden** (as expected).
- Accessing /api/profiles/ without login returned **403 Forbidden** (as expected).

5. Screenshots



- Admin panel showing users and models.

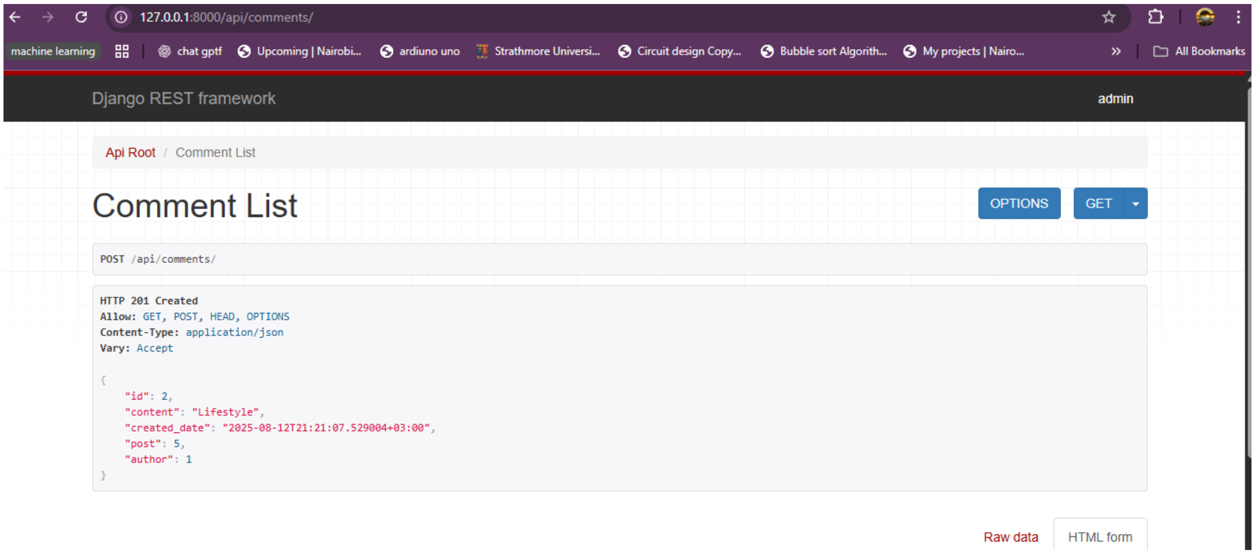


```
POST /api/posts/
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 5,
    "title": "Lifestyle",
    "slug": "Lifestyle",
    "content": "Life",
    "published_date": "2025-08-12T21:19:27.408789+03:00",
    "updated_date": "2025-08-12T21:19:27.408837+03:00",
    "author": 1,
    "categories": [
        4
    ]
}
```

- Successful GET request in DRF for /api/posts/.



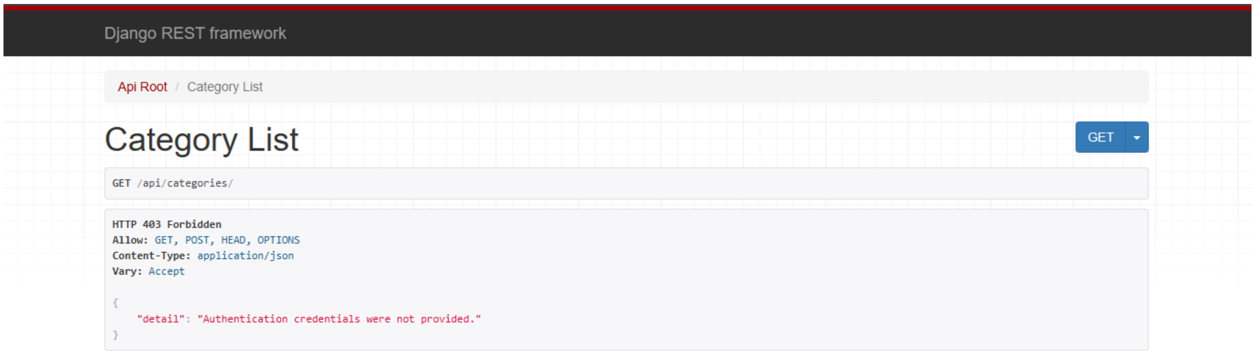
Comment List

```
POST /api/comments/
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 2,
    "content": "Lifestyle",
    "created_date": "2025-08-12T21:21:07.529004+03:00",
    "post": 5,
    "author": 1
}
```

- Successful POST request for /api/comments/.



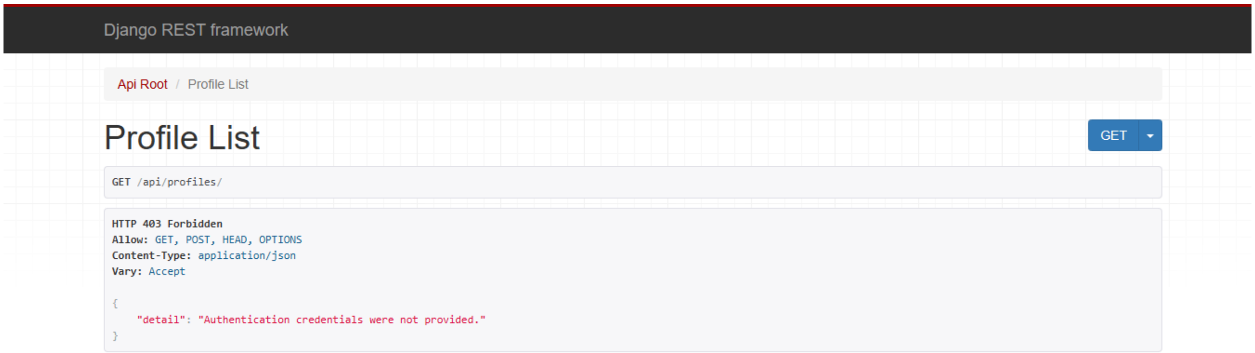
Category List

```
GET /api/categories/
```

```
HTTP 403 Forbidden
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "Authentication credentials were not provided."
}
```

- Forbidden (403) error for unauthorized access to /api/categories/.



Profile List

```
GET /api/profiles/
```

```
HTTP 403 Forbidden
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "Authentication credentials were not provided."
}
```

- Forbidden (403) error for unauthorized access to /api/profiles/.

6. Challenges & Solutions

- **Issue:** NameError: name 'admin' is not defined in urls.py
  Solution: Imported from django.contrib import admin.
- **Issue:** 403 Forbidden for /api/profiles/
  Solution: Logged in via DRF's login option or Django admin before making requests.
- **Issue:** Permission mismatch for categories.
  Solution: Used IsAdminUser permission class in CategoryViewSet.

7. Conclusion

The Django Blogsite API is now fully functional with authentication, permissions, and CRUD operations for all key resources. Permissions work as intended, restricting sensitive actions to authorized users only.