

Spam Detection API

Ananya Tripathi

20/07/2024

Overview

Welcome to the Spam Detection API! This project is a Django-based web application designed to manage user profiles, contacts, and spam records. It includes features for user registration, login, profile management, contact management, spam marking, and search functionality.

Project Structure

The project consists of the following six Django apps:

1. **accounts**: Manages user registration, authentication, and profile operations.
2. **search**: Provides functionality to search for users, contacts, and spam records.
3. **contacts**: Handles operations related to user contacts.
4. **spam_management**: Manages the marking and tracking of spam numbers.
5. **user_profile**: Manages user profile details.
6. **management_utils**: Contains custom management commands for tasks like data population.

Test Cases Overview

Test cases are implemented to validate the functionality of various API endpoints and ensure that the application behaves as expected. The following areas are covered in the test cases:

1. User Management: Tests for user registration, login, and profile updates.
2. Contact Management: Tests for adding, retrieving, and deleting contacts.
3. Spam Management: Tests for marking numbers as spam and retrieving spam counts.
4. Search Functionality: Tests for searching users and contacts by name and phone number

Data Population

The `management_utils` app includes a custom management command for populating the database with sample data. This command will:

- Create 50 users.
- Add 5-10 contacts for each user.
- Mark 10 users and 20 contacts as spam.
- Add 10 random phone numbers to the spam table.
- Ensure 5 numbers are present in all three tables (users, contacts, spam).

Key Learnings

1. Custom Management Commands: Gained experience in creating and using custom management commands for data population and other tasks.
2. Data Modeling: Improved understanding of Django's ORM and relational database management through practical implementation of models and relationships.
3. API Testing: Enhanced skills in writing and executing test cases to ensure the reliability and correctness of API endpoints.

Running the Application

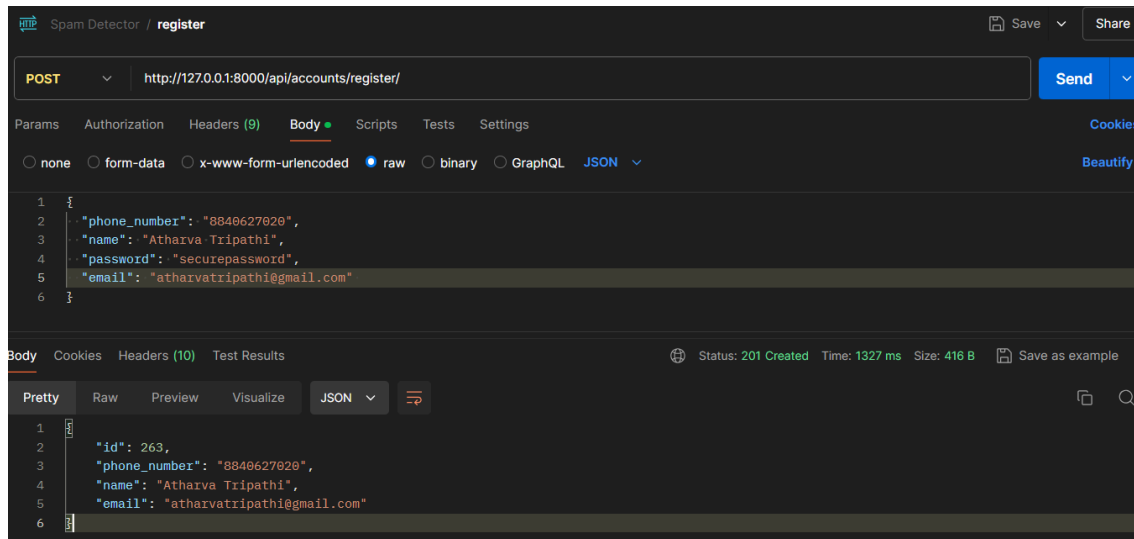
To run the application and explore its features, please refer to the `README.md` file for detailed instructions on installation, migration, and server startup.

API Endpoints

1. User Management

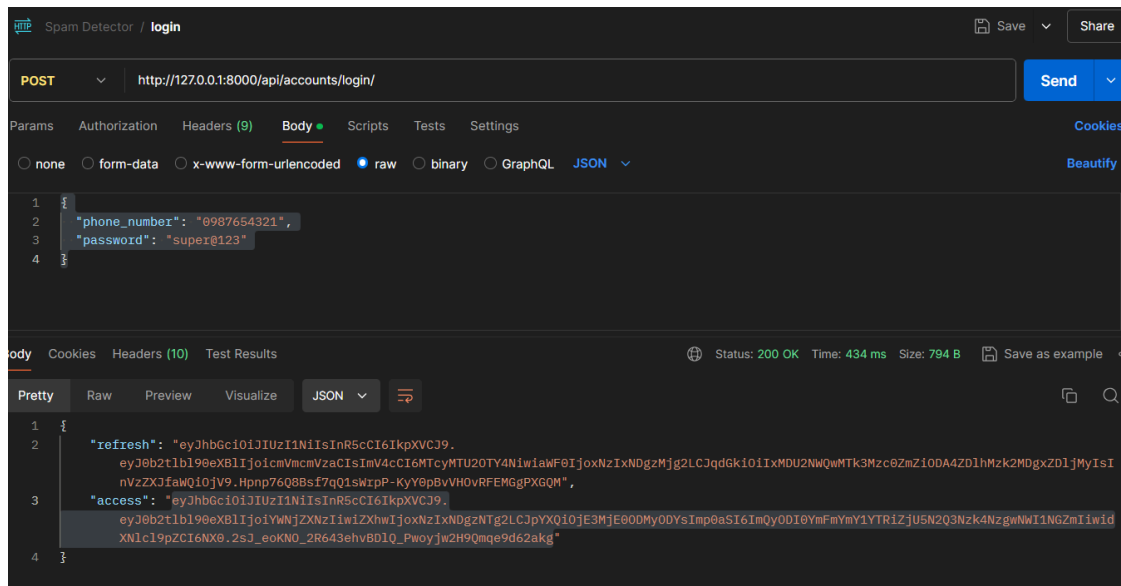
Register User

- **Endpoint:** `POST /api/accounts/register/`



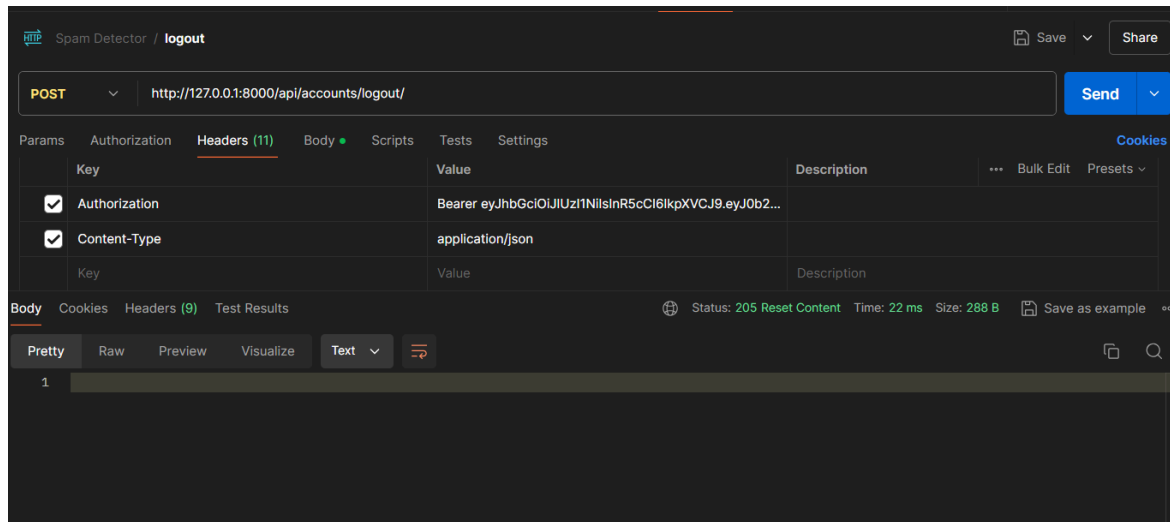
Login

- **Endpoint:** POST /api/accounts/login/



Logout

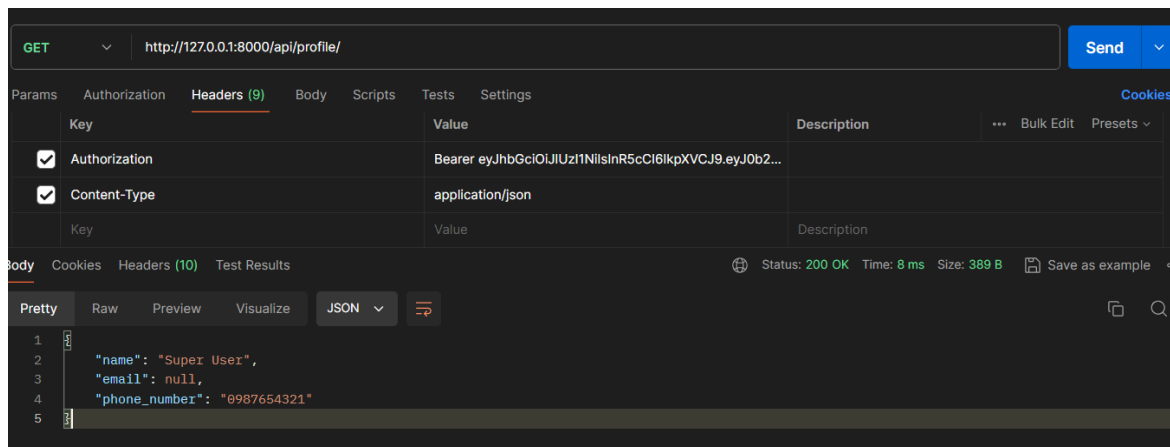
- **Endpoint:** POST /api/accounts/logout/



2. Profile Management

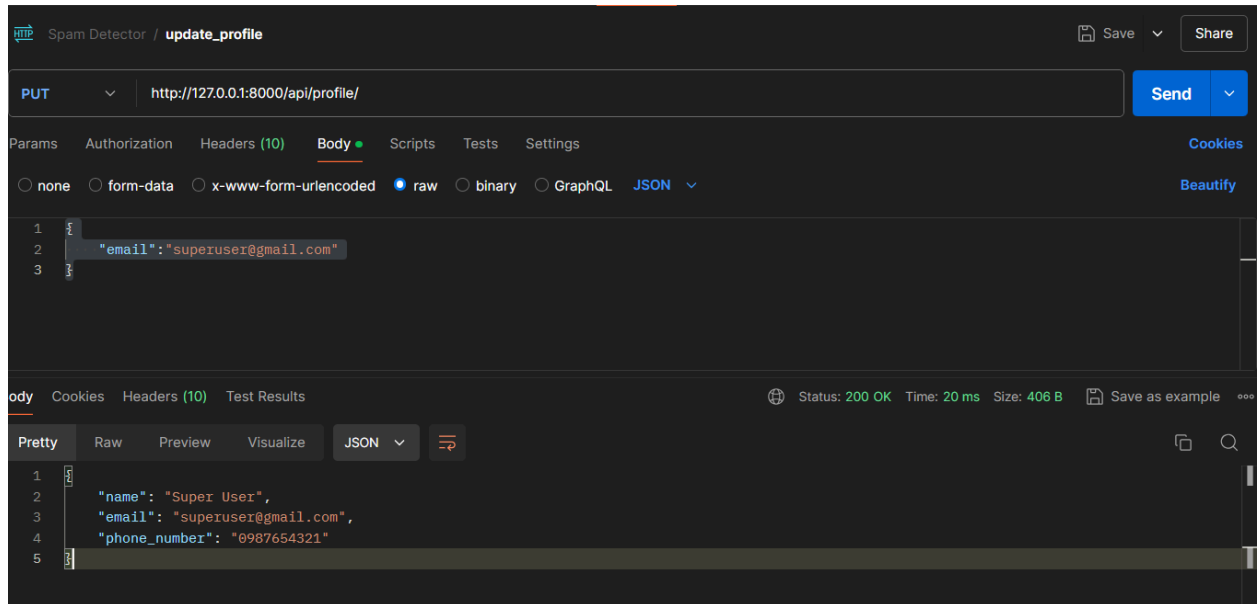
Get Profile

- Endpoint: GET /api/profile/



Update Profile

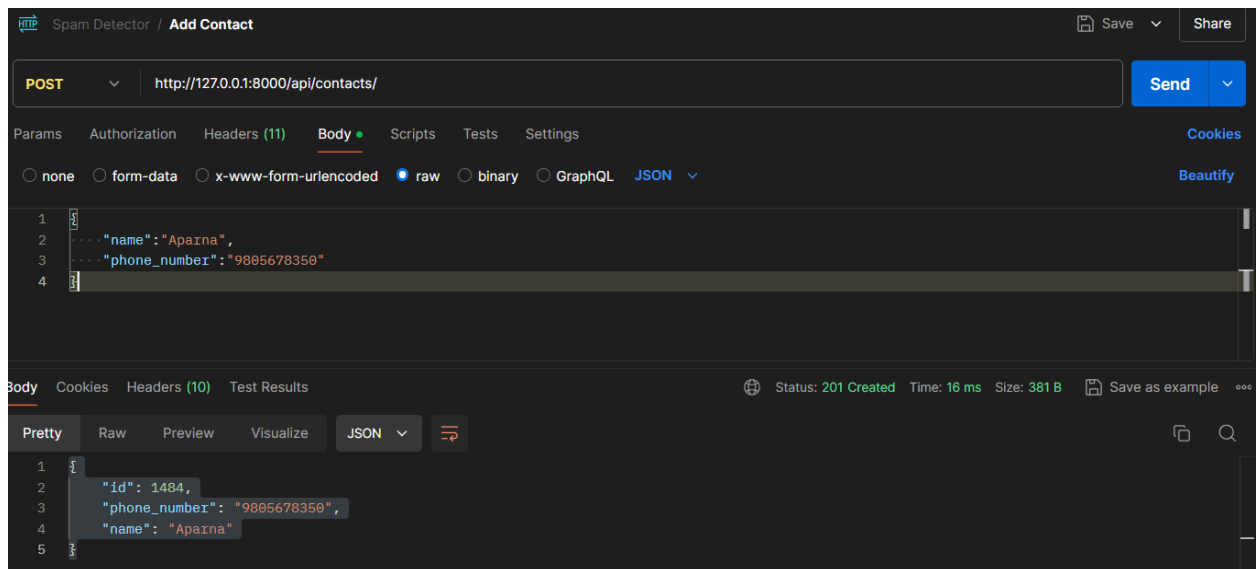
- Endpoint: PUT /api/profile/



3. Contact Management

Add Contact

- **Endpoint:** POST /api/contacts/



Get Contacts

- **Endpoint:** GET /api/contacts/

Spam Detector / **Get Contact** Save Share

GET ▼ Send ▼

Params Authorization **Headers (9)** Body Scripts Tests Settings Cookies

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Body Cookies Headers (10) Test Results Status: 200 OK Time: 7 ms Size: 378 B Save as example

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 [
2   {
3     "id": 1484,
4     "phone_number": "9885678358",
5     "name": "Aparna"
6   }
7 ]
```

Delete Contact

- Endpoint: **DELETE** /api/contacts/<contact_id>

Spam Detector / **Delete Contact** Save Share

DELETE ▼ Send ▼

Params Authorization **Headers (9)** Body Scripts Tests Settings Cookies

Headers 7 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Body Cookies Headers (9) Test Results Status: 204 No Content Time: 22 ms Size: 287 B Save as example

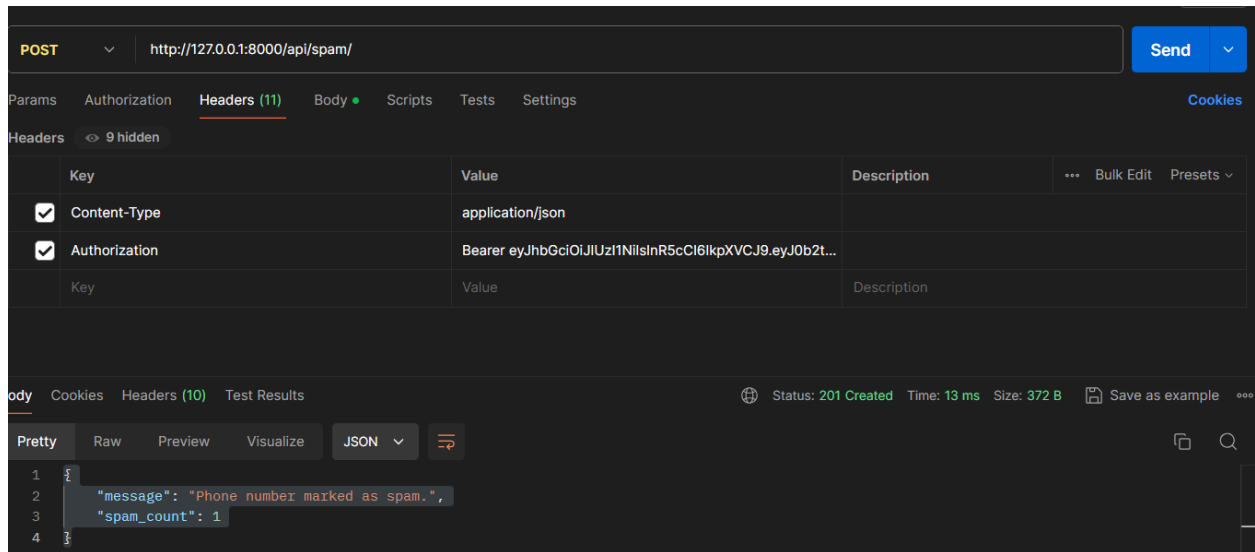
Pretty Raw Preview Visualize **Text** ▼ ≡

```
1
```

4. Spam Management

Mark Spam

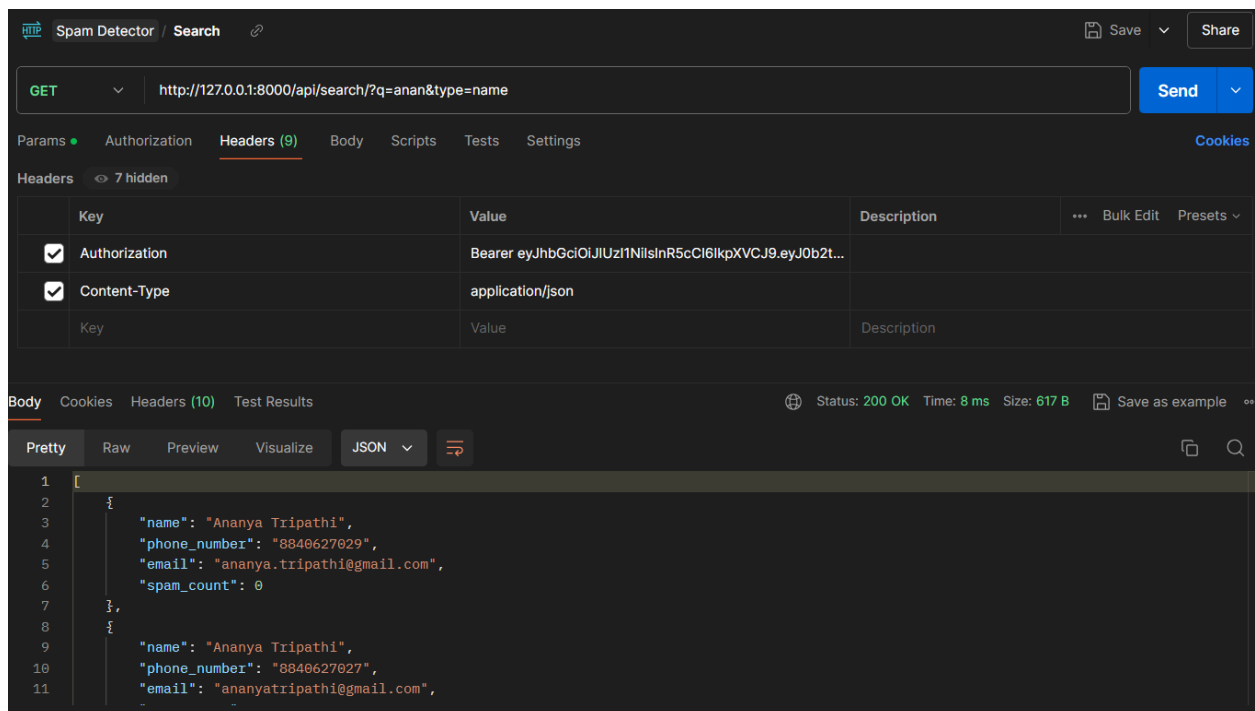
- Endpoint: **POST** /api/spam/



5. Search

Search by Name

- Endpoint: GET /api/search/?q=<name>&type=name



Search by Phone

- Endpoint: GET /api/search/?q=<phone_number>&type=phone

GET

http://127.0.0.1:8000/api/search/?q=567&type=phone

Send

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Cookies

Headers

7 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

body

Cookies

Headers (10)

Test Results

Status: 200 OK

Time: 31 ms

Size: 4.49 KB

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "name": "User0",
4      "phone_number": "123456780",
5      "email": null,
6      "spam_count": 0
7    },
8    {
9      "name": "User1",
10     "phone_number": "123456781",
11     "email": null,
```