

Instahyre Assessment

Submitted By: Rohan Singh

Phone Number: 9380461970

The project contains the following functionality:

1. Backend REST API using Django Framework for mobile application.
2. 5 Rest API for Registration, Login, Search by Name, Search by Phone number and marking a phone number as spam.
3. Technologies used: Python, Django, ORM [Relational Database]

How to run the code:

- Download the zip file and extract.
- Ensure Django and “djangoestframework” are installed.
- Run “python manage.py makemigrations” and “python manage.py migrate”.
- Run “python manage.py runserver”.
- The server will be up and running.
- Test the APIs using Postman/ARC etc.

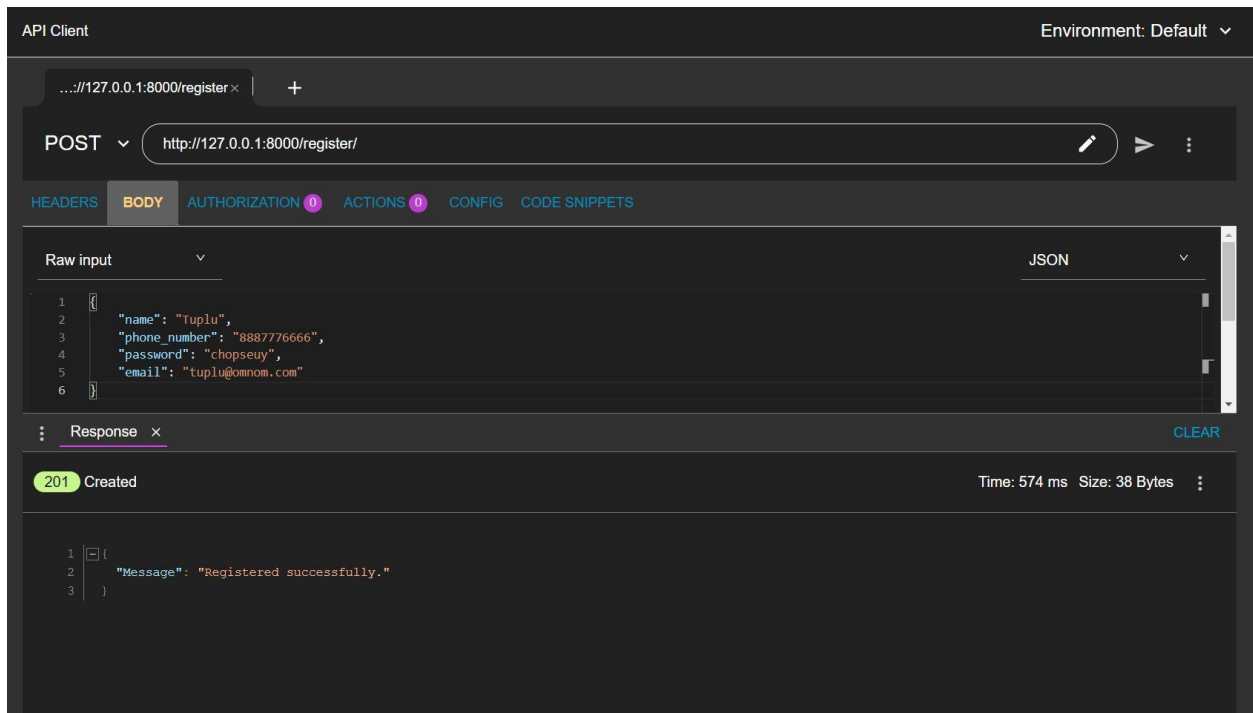
Factors considered for Production level API:

1. Security: Implemented Django’s authentication mechanism for searching and marking a number as spam.
2. Scalability and Modularity: Created a utility file profile_services.py to handle logic, segregated from views.py.
3. Usage of ORM: Utilized Django’s models.py class instead of raw SQL queries.
4. Followed SOLID principles, ACID principles, and Design Patterns.
5. REST API Best Practices:
 - JSON format instead of plain texts.
 - Correct usage of HTTPS Status code: 200 for success, 401 for unauthorized, etc.
 - Plural nouns for resources.
 - Usage of query string.

Further scope of improvement: Rate limiting and throttling.

Snapshots of the API Testing:

1. Register:



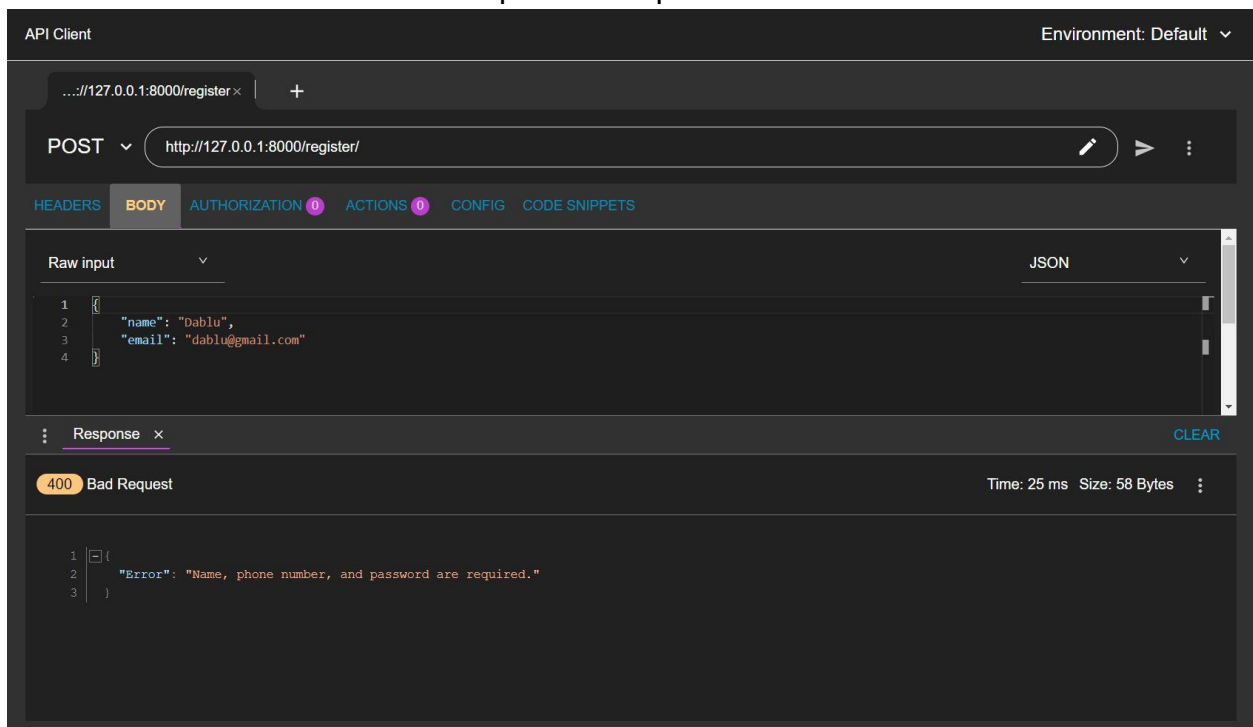
The screenshot shows the API Client interface with the following details:

- Environment:** Default
- Request Method:** POST
- Request URL:** http://127.0.0.1:8000/register/
- Request Body (JSON):**

```
{  "name": "Tuplu",  "phone_number": "8887776666",  "password": "chopseuy",  "email": "tuplu@omnom.com"}
```
- Response Status:** 201 Created
- Response Time:** 574 ms
- Response Size:** 38 Bytes
- Response Body (JSON):**

```
{  "Message": "Registered successfully."}
```

When we don't enter username/password/phone number:



The screenshot shows the API Client interface with the following details:

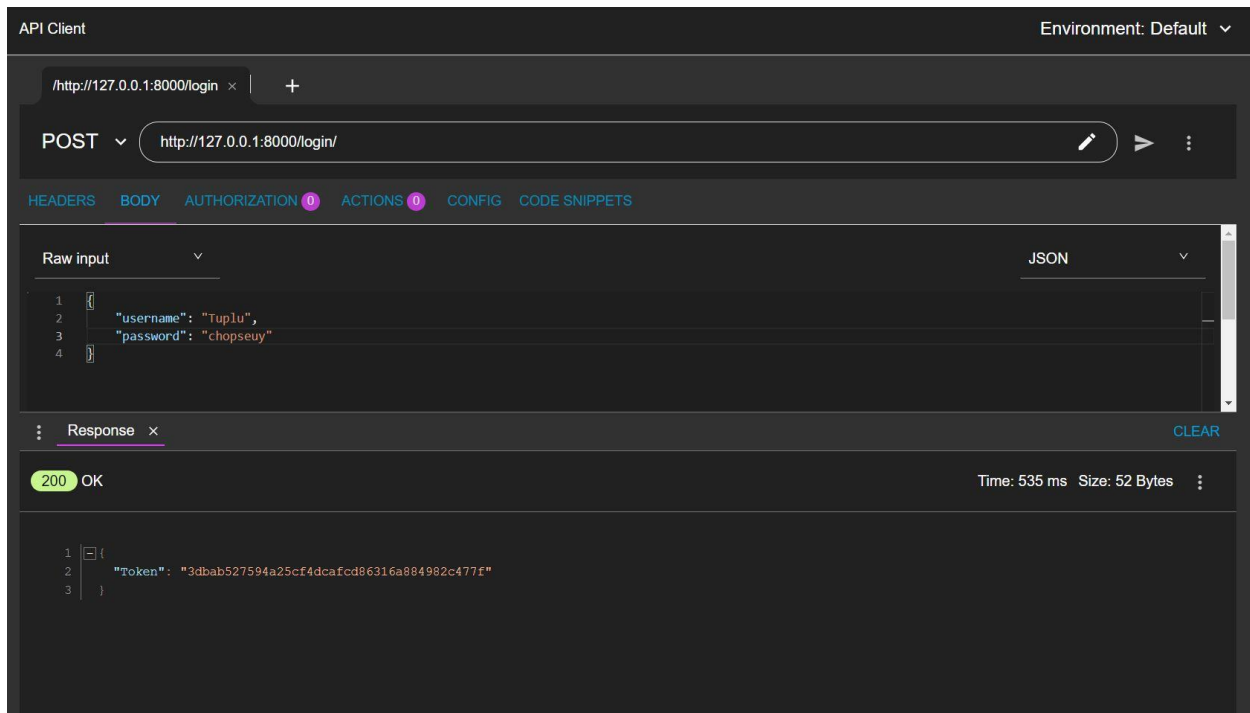
- Environment:** Default
- Request Method:** POST
- Request URL:** http://127.0.0.1:8000/register/
- Request Body (JSON):**

```
{  "name": "Dablu",  "email": "dablu@gmail.com"}
```
- Response Status:** 400 Bad Request
- Response Time:** 25 ms
- Response Size:** 58 Bytes
- Response Body (JSON):**

```
{  "Error": "Name, phone number, and password are required."}
```

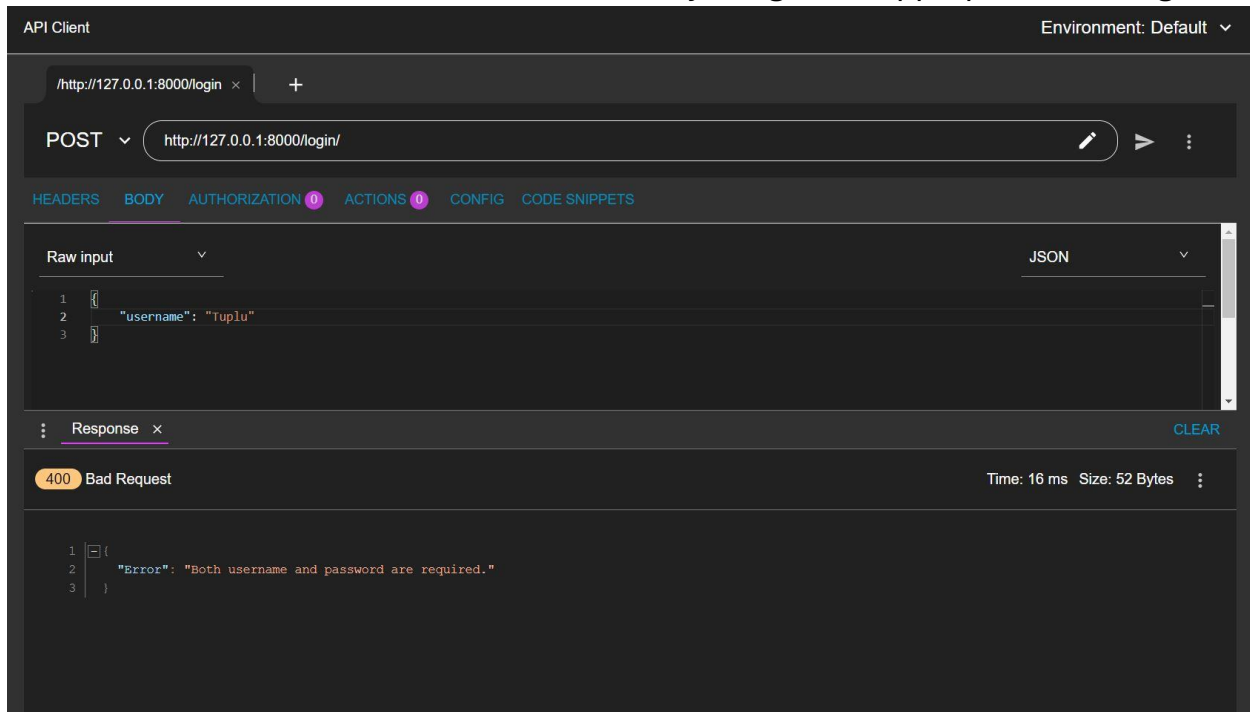
2. Login

Upon successful login, we get the authentication token in response along with status code 200



The screenshot shows the API Client interface. The top bar indicates the environment is 'Default'. The URL bar shows the endpoint `/http://127.0.0.1:8000/login`. The method is set to **POST**. The request body is in JSON format, containing `{ "username": "Tuplu", "password": "chopseuy" }`. The response is shown below, with a status of **200 OK**, a time of 535 ms, and a size of 52 Bytes. The response body is `{ "Token": "3dbab527594a25cf4dcafc86316a884982c477f" }`.

When we don't enter the credentials correctly, we get the appropriate message:

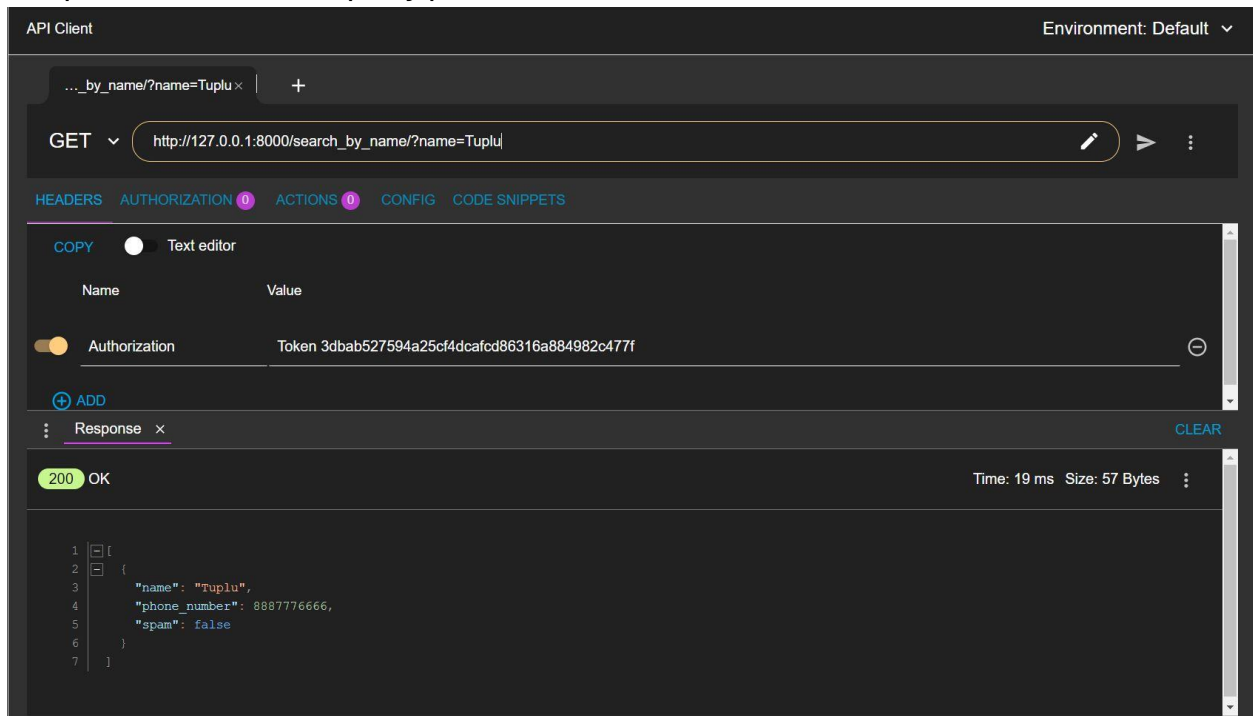


The screenshot shows the API Client interface. The top bar indicates the environment is 'Default'. The URL bar shows the endpoint `/http://127.0.0.1:8000/login`. The method is set to **POST**. The request body is in JSON format, containing `{ "username": "Tuplu" }`. The response is shown below, with a status of **400 Bad Request**, a time of 16 ms, and a size of 52 Bytes. The response body is `{ "Error": "Both username and password are required." }`.

3. Search by Name

We provide the token in the header for authentication.

We pass the name as query parameter.



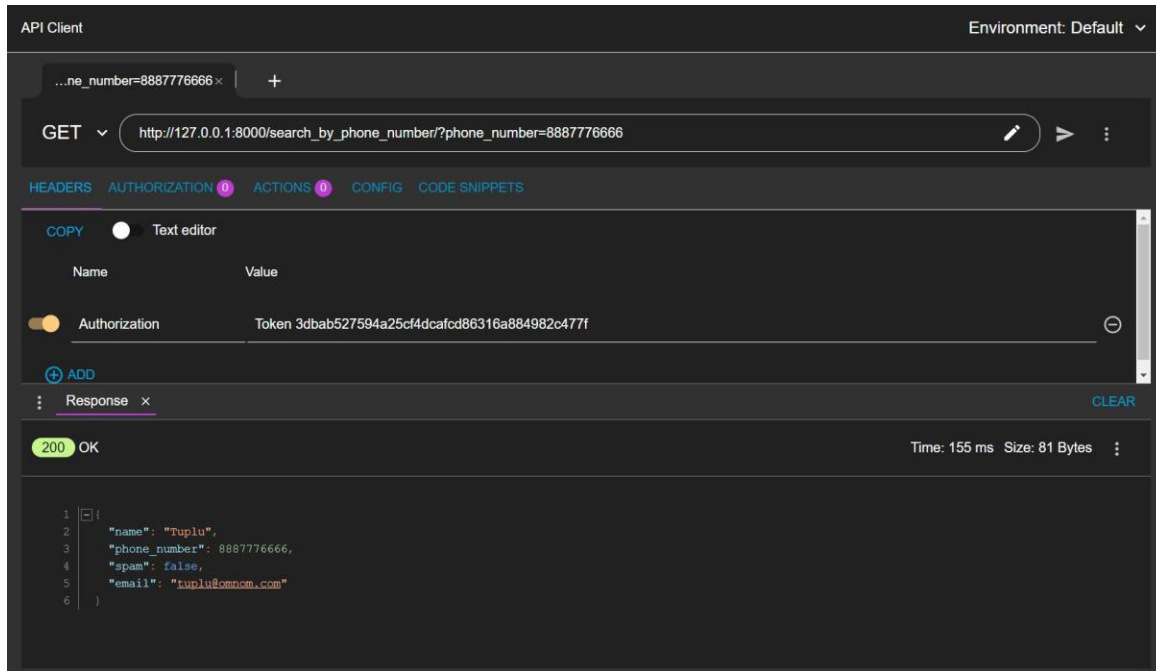
In response we can see whether the name is spam or not.

Appropriate messages are shown if the details do not exist.

4. Search by Number

We provide the token in the header for authentication

We pass the phone number as a query parameter.



In response we can see whether the number is spam or not.

Appropriate messages are shown if the details do not exist.

5. Marking a number as spam

We provide the token in the header

We provide the phone number in body as a JSON.

The screenshot shows the API Client interface with the following details:

- Environment:** Default
- Request Method:** POST
- Request URL:** http://127.0.0.1:8000/spams/
- Headers:**
 - Content-Type: application/json
 - Authorization: Token 464ad79c3e10dc741032cb3f0f67a77839bd32dc
- Response:** 200 OK, Time: 22 ms, Size: 50 Bytes
- Response Body (JSON):**

```
1 {
2   "Message": "Contact successfully marked as spam."
3 }
```

The screenshot shows the API Client interface with the following details:

- Environment:** Default
- Request Method:** POST
- Request URL:** http://127.0.0.1:8000/spams/
- Body:**

```
1 {
2   "phone_number": "9993334444"
3 }
```
- Response:** 200 OK, Time: 22 ms, Size: 50 Bytes
- Response Body (JSON):**

```
1 {
2   "Message": "Contact successfully marked as spam."
3 }
```