

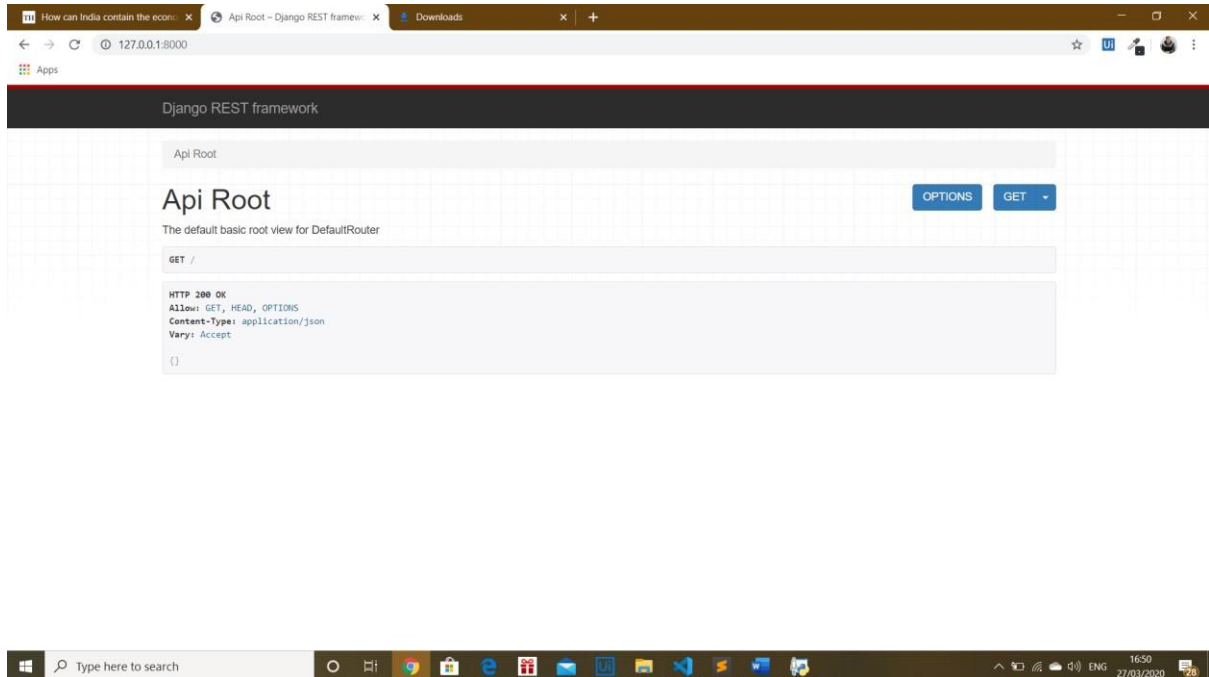
Login API Documentation

For Running server

1. Extract the following zip file in a suitable folder
2. To run following code run the application drive into the extracted path and hit command "manage.py runserver" on cmd

Open your browser and enter <http://127.0.0.1:8000/>

A welcome page will get opened showing server is running



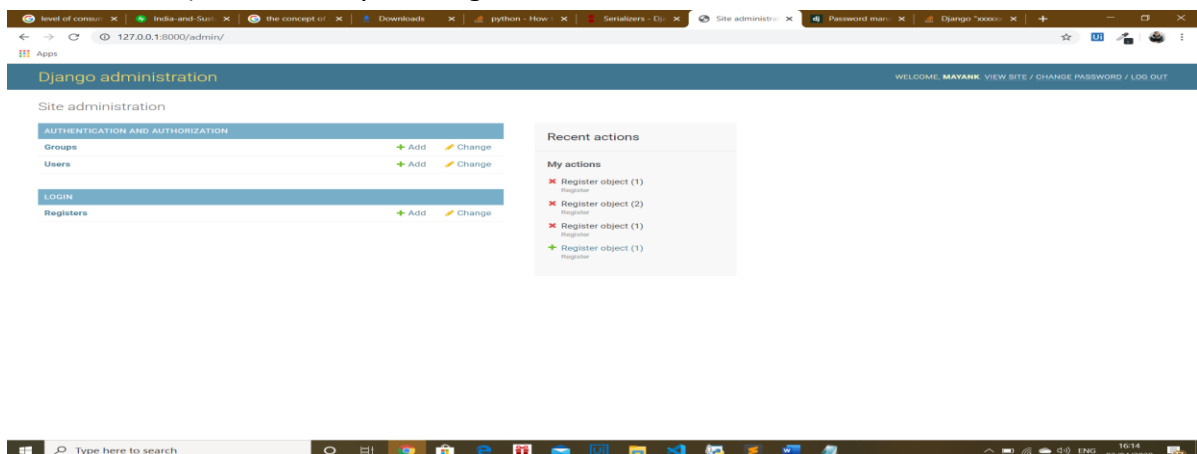
Admin details are provided below

Open

<http://127.0.0.1:8000/admin/>

Admin username: mayank

Admin password: mayankkungani



API Documentation:

Allowed HTTPs requests:

POST :: <http://127.0.0.1:8000/api/register/>

GET : <http://127.0.0.1:8000/api/login/>

1. There are two API endpoints provided whose url are
 - a. <http://127.0.0.1:8000/api/login/>
 - b. <http://127.0.0.1:8000/api/register/>
2. For calling API use can use a dedicated software **"POSTMAN"** or you can use python libraries request for sending GET and POST requests

Let us see how **REGISTER** endpoint works :

It allows user registration using POST request.

User attributes:

- name (String) : Full Name.
- email (String) : email id of the user
- phone(int) :. Mob. No. of the user
- userid(Strind): unique user id
- pass1: password of user
- passconf: password conformation

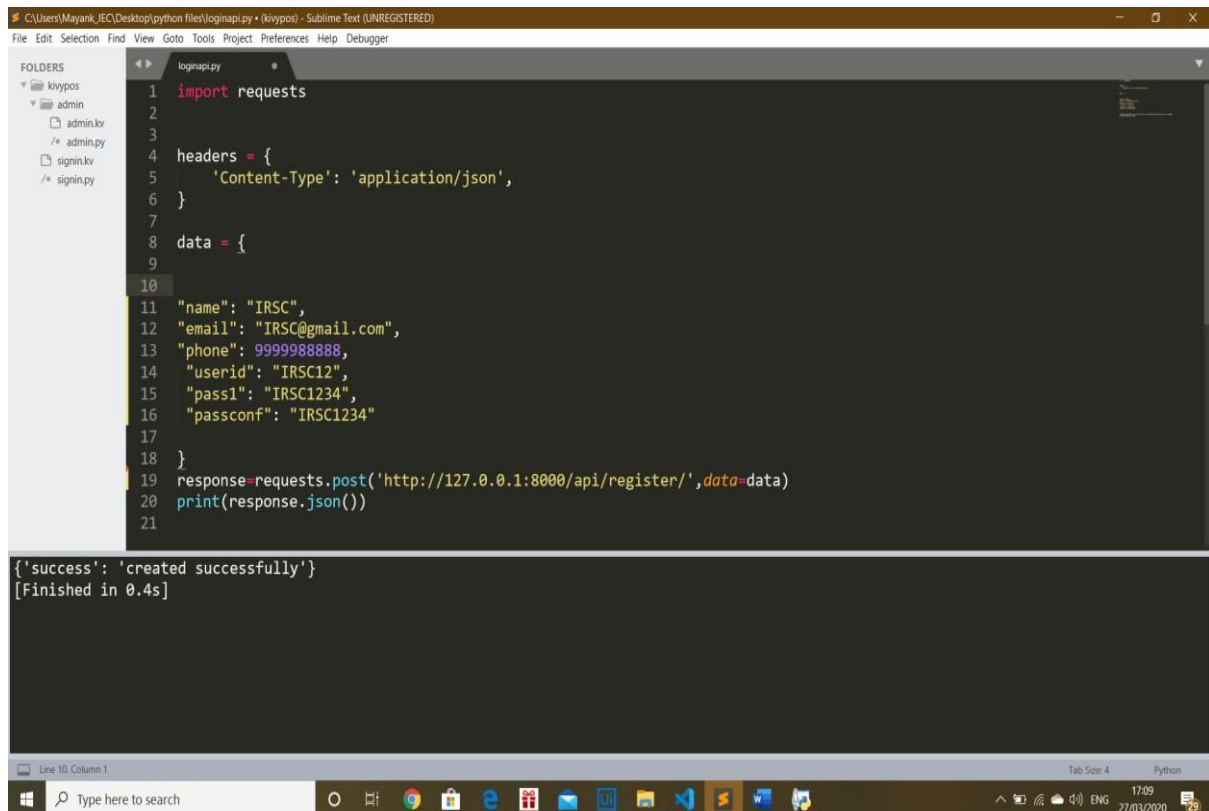
Sample code is shown below

Open your ide and write the following code

```
'''  
import requests  
headers = {  
    'Content-Type': 'application/json',  
}
```

```
data = {
    "name": "IRSC",
    "email":
    : "IRSC@gmail.com",
    "phone": 9999988888,
    "userid": "IRSC12",
    "pass1": "IRSC1234",
    "passconf": "IRSC1234"}
response=requests.post('http://127.0.0.1:8000/api/register/',data=data)
print(response.json())
```

'''



```
C:\Users\Mayank\IEC\Desktop\python files\loginapi.py • (kivypos) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help Debugger

FOLDERS
  kivypos
    admin
      admin.kv
      /* admin.py
    signin.kv
    /* signin.py

loginapi.py
1  import requests
2
3
4  headers = {
5      'Content-Type': 'application/json',
6  }
7
8  data = {
9
10
11  "name": "IRSC",
12  "email": "IRSC@gmail.com",
13  "phone": 9999988888,
14  "userid": "IRSC12",
15  "pass1": "IRSC1234",
16  "passconf": "IRSC1234"
17  }
18
19  response=requests.post('http://127.0.0.1:8000/api/register/',data=data)
20  print(response.json())
21

{'success': 'created successfully'}
[Finished in 0.4s]
```

A success message will come indicating that you Registered successfully.

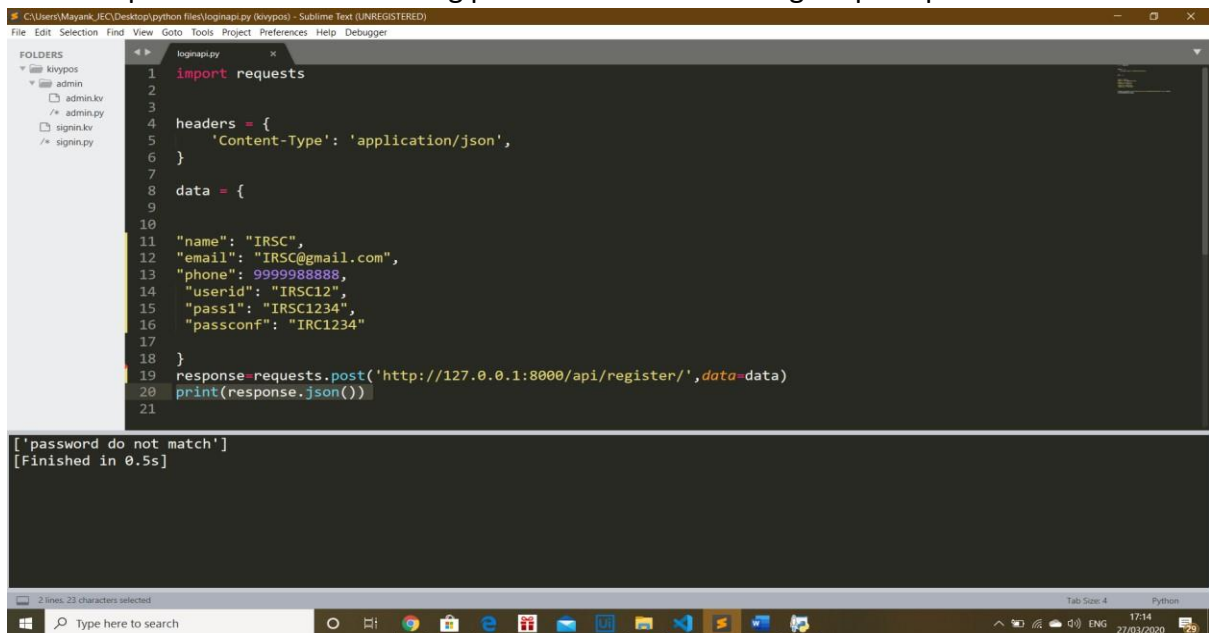
The parameters that have been checked before registering a user are:

- 1.Email must be Unique and a valid one.
2. Userid must be unique.
- 3.The two passwords should match.
4. The password must be of minimum 8 characters.

5.The passwords should save in a database in a hashed code , it must be hidden.

If any of the about field is invalid the server returns the corresponding message

For Example here an error showing passwords do not message is prompted



```
1 import requests
2
3
4 headers = {
5     'Content-Type': 'application/json',
6 }
7
8 data = {
9
10
11 "name": "IRSC",
12 "email": "IRSC@gmail.com",
13 "phone": "999988888",
14 "userid": "IRSC12",
15 "pass1": "IRSC1234",
16 "passconf": "IRC1234"
17
18 }
19 response=requests.post('http://127.0.0.1:8000/api/register/',data=data)
20 print(response.json())
21
```

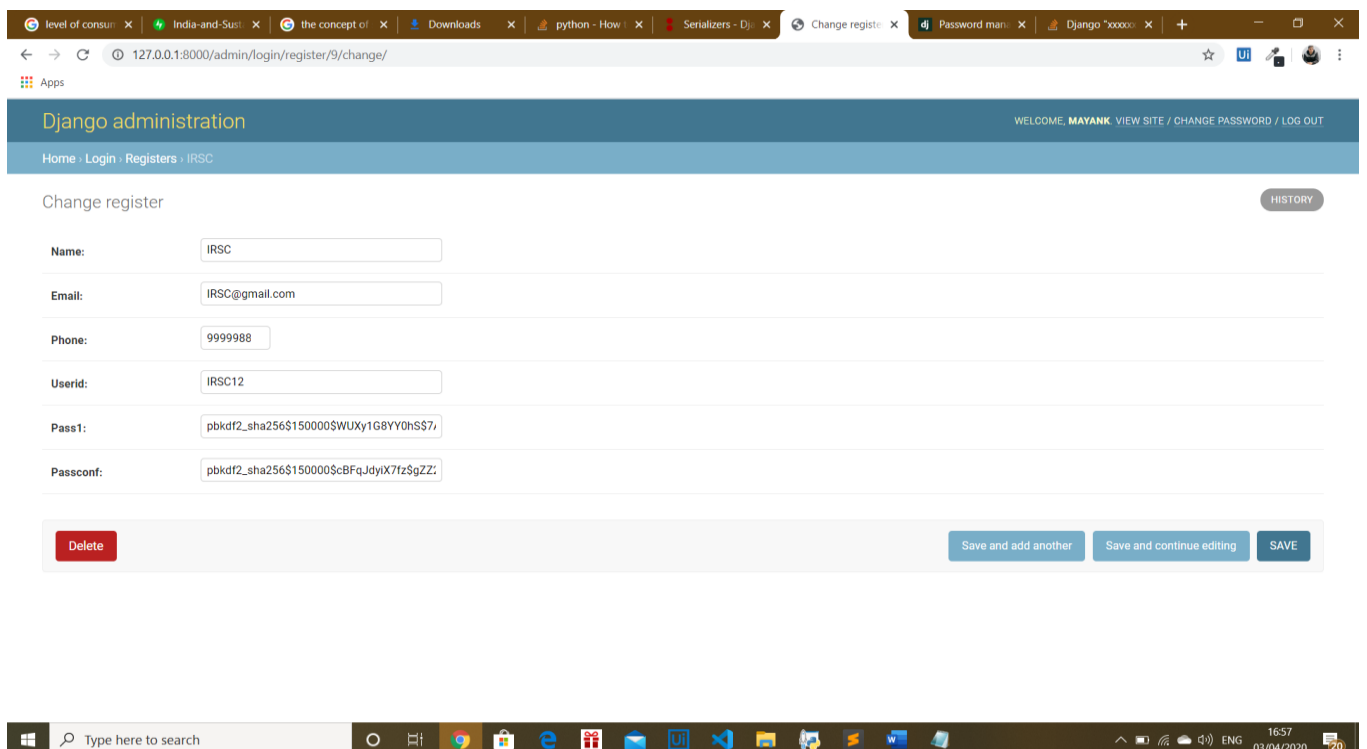
['password do not match']
[Finished in 0.5s]

Note: The passwords are getting HASHED before saving in the database . this can be seen from admin site i.e <http://127.0.0.1:8000/admin/>

Admin username: mayank

Admin password: mayankkungani

Sample Screenshot is provided below see fields **pass1** and **passconf**



Django administration

WELCOME, MAYANK VIEW SITE / CHANGE PASSWORD / LOG OUT

Home Login Registers IRSC

Change register HISTORY

Name: IRSC

Email: IRSC@gmail.com

Phone: 9999888

Userid: IRSC12

Pass1: pbkdf2_sha256\$15000\$WUXy1G8YY0hS\$7/

Passconf: pbkdf2_sha256\$15000\$ScBfQJdyX7z\$gZZ;

Delete Save and add another Save and continue editing SAVE

Now coming to Login endpoint

GET : <http://127.0.0.1:8000/api/login/>

1. It allows GET requests and provides all biofields if authentication matches i.e ***userid and password***
2. ***Copy the code below*** in your IDE

```
'''
```

```
import requests
```

```
headers = {
```

```
    'Content-Type': 'application/json',
```

```
}
```

```
data = {
```

```
    "userid": "IRSC12",
```

```
    "pass1": "IRSC1234",
```

```
}
```

```
response=requests.get('http://127.0.0.1:8000/api/login/',data=data)
```

```
print(response.json())
```

```
'''
```

```
1 import requests
2 headers = {
3     'Content-Type': 'application/json',
4 }
5 data = {
6     "userid": "IRSC12",
7     "pass1": "IRSC1234"}
8
9
10 response=requests.get('http://127.0.0.1:8000/api/login/',data=data)
11 print(response.json())
12
13
```

```
{"name": "IRSC", "phone": "9999988888", "email": "IRSC@gmail.com", "pass1": "pbkdf2_sha256$150000$WUXy1G8Y0hs$7A9+8y1uW9jnQLKhb5Qjvs2gS9IIo+T9LSDahfiHkmg="}
[Finished in 1.3s]
```

If authentication is valid a valid message showing all BIO-Fields like name ,phone and email.

The following code takes care for the following :

- 1.Returns a failed message if userid and password do not match.
2. Return a response if userid and password is empty.

Sample is provided below

```
1 import requests
2
3
4 headers = {
5     'Content-Type': 'application/json',
6 }
7
8 data = {
9
10     "userid": "IRSC12",
11     "pass1": "IRSC1234",
12 }
13
14
15 response=requests.get('http://127.0.0.1:8000/api/login/',data=data)
16 print(response.json())
17
```

```
['incorrect password']
[Finished in 0.4s]
```

THANK U , Hoping for a positive reply

