

CODE GRAPGERS PYTHON TEST TASK

Create a flask API using flask-restful and Postgres DB.

API-Endpoints:

User APIs: -

- **Register User: -**

<http://127.0.0.1:5000/signUp>

It is used to register new users. Variable required for signUp first name (required), last name, email (required, unique and email validation), username (required and unique) password (required and at least 10 characters validation).

- **SigIn User: -**

<http://127.0.0.1:5000/signIn>

It is used for login into the system by using basic Authentication method and generate jwt and refresh token to the user to access the remaining api's. Variable required username and password.

- **Refresh Token: -**

<http://127.0.0.1:5000/refresh>

It is used to refresh the jwt token when it expires (1 hour). So, user not need to login again and again in a day. Refresh token is set in cookies after signIn into the system. And refresh token will expire after 1 day.

Jobs APIs: -

You need the jwt token to access the below apis.

- **Add Job: -**

<http://127.0.0.1:5000/jobs>

It is used to add new jobs. Variable are job title, job description, job rate, latitude, longitude, is active(for soft delete), user_id (from jwt_token), job created, job updated.

- **Update Job: -**

<http://127.0.0.1:5000/jobs/6>

It is used to update the old jobs by its ID. It could be full update or partial update.

- **Soft Delete Job: -**

<http://127.0.0.1:5000/jobs/4>

It is used to delete a job by its ID. It would change the "is active" flag in the DB.

- **Hard Delete Job: -**

<http://127.0.0.1:5000/jobs/del/4>

It is used to complete delete of the job from DB by its ID.

- **Get Jobs: -**

<http://127.0.0.1:5000/jobs>

It is used to get all jobs from the DB.

<http://127.0.0.1:5000/jobs?id=2>

It is used to filter by job ID.

<http://127.0.0.1:5000/jobs?lat=32.3915&longi=74.4342&kilometer=15>

It is used to filter jobs by user's location and its radius given by users lat, long and kilometer.

Swagger Documentation: -

I also made swagger documentation of that API. You can access it from that:

<http://localhost:5000/apidocs/>

- **Screenshot:-**

The screenshot displays the Swagger UI for an API named 'apispec_1.json'. The interface includes a top navigation bar with the Swagger logo and an 'Explore' button. Below the header, the API title 'A swagger API' is shown with version '0.0.1'. The main content area lists several endpoints under the 'default' tab:

- GET /jobs**: All Jobs.
- POST /jobs**: Add Job.
- DELETE /jobs/del/{id}**: Hard Delete Job.
- DELETE /jobs/{id}**: Soft Delete Job.
- PUT /jobs/{id}**: Update Job.
- POST /refresh**: Refresh Token.
- POST /signIn**: User Signin. (Marked as secure with a lock icon)
- POST /signUp**: Create New User.

At the bottom, there is a 'Models' section showing two data models: 'job' and 'user', each with a right-pointing arrow indicating further details.

[Powered by [Flasgger](#) 0.9.5]

Unit Testing: -

I also write the basic unit testing of the above APIs. You can see that in the [github repo](#).

Dumi API Frontend: -

I also write the Dumi frontend of the above APIs. You can see that in the [github repo](#).

Postgres DB: -

Screenshot: -

The screenshot shows the pgAdmin interface with the 'users' table selected in the 'public' schema. The Query Editor displays the following SQL query:

```
1 SELECT * FROM public.users
2 ORDER BY id ASC
```

The Data Output tab shows the following data:

id	fName	lName	email	userName	password
1	akhlaq	mahar	abc@gmail.com	akhlaq	sha256\$zw4jTz1grAZfEYjP\$acea8f
2	ali	ahmad	abc2@gmail.com	ali	sha256\$GbZQDbasDOK55dC6\$dbc
3	akhlaq	mahar	abc12@gmail.com	akhlaq1	sha256\$b512PNAF7JO873SD\$549
4	akhlaq	mahar	abc1234@gmail.com	akhlaq3	sha256\$Sn9x9VeHda2mcd7QI\$966

The screenshot shows the pgAdmin interface with the 'jobs' table selected in the 'public' schema. The Query Editor displays the following SQL query:

```
1 SELECT * FROM public.jobs
2 ORDER BY id ASC
```

The Data Output tab shows the following data:

id	user_id	jobTitle	jobDesc	jobRate	latitude	long
1	1	ML Engineer	Lahore	100k	31.5204	74.3
2	2	Senior Software Engineer	Saudi Arabia	30k	32.3363	74.3
3	3	Software Engineer	Sialkot	30k	32.4945	74.3
4	5	ML	Multan	50k	30	77
5	6	ML	Karachi	30k	31.5204	74.3
6	7	IT Support	Pakistan	30k	30	77
7	8	Software Engineer	Pakistan	30k	31.5204	74.3
8	9	Software Engineer	Pakistan	30k	31.5204	74.3
9	10	Software Engineer	Pakistan	30k	31.5204	74.3
10	11	Software Engineer	Pakistan	30k	31.5204	74.3
11	17	Software Engineer	multan	30k	31.5204	74.3
12	18	Software Engineer	multan	30k	31.5204	74.3

GitHub Repo Link: -

<https://github.com/Muhammad-Akhlaq/Jobs-API-Using-Flask-And-Postgres>

Regards,

Muhammad Akhlaq

m.akhlaq.mahar123@gmail.com

<https://www.linkedin.com/in/muhammad-akhlaq-4a7b9b155/>

03316631105