

Hiring-App

By Almog Lev

Overview:

Hiring-App is a Django app that finds the best matching candidates for a given job. It takes into consideration the title of the job – **full title match and partial title match** and the match between the job required skills and the candidates' skills.

Extras Features and Functionalities:

1. Partial title match
2. Taking performance into consideration by adding 2 constraints:
 - a. `MIN_THRESHOLD` - Threshold to determine what is a "best candidate" - must meet at least % of the required skills of the job
 - b. `MAX_CANDIDATES_ABOVE_THRESHOLD` - When reaching more than this number we have enough best candidates so we can stop
3. Unit tests
4. Admin Panel
5. API Overview that includes all endpoints – the required one + 3 GET endpoints (one for every model), can be seen in the URL (after running the server):

<http://localhost:8000/api/>

GET /api/

HTTP 200 OK

Allow: GET, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "API Overview": "http://localhost:8000/api",
  "Admin Panel": "http://localhost:8000/admin",
  "GET best candidates": "http://localhost:8000/api/best-candidate-list/<int:jobId>",
  "GET all skills": "http://localhost:8000/api/skill-list",
  "GET all candidates": "http://localhost:8000/api/candidate-list",
  "GET all jobs": "http://localhost:8000/api/job-list"
}
```

Running locally:

Assuming virtualenv is installed and activated, django app is already created and mysqlclient is already created (follow these instructions to do so:

https://docs.google.com/document/d/1gCKuA5EWBn-Ayp3m_SuMt5Ot7dKkPF12o2urhPNy6dU/edit).

Instructions:

DB:

- After activating mysqlclient via XAMPP you can explore the database here:
<http://localhost/phpmyadmin/>
- Create migrations:
`python manage.py makemigrations`
- Migrate:
`python manage.py migrate`
- Two ways to populate the matcherdb:
 1. Attached to the email 3 files named: api_skill.sql, api_candidate.sql, api_job.sql
They contain all the needed queries (creating the tables + populating them)
 2. Populate manually in the Admin Panel (As seen below)

Server:

- Run the server:
`python manage.py runserver`
- Go to <http://localhost:8000/api/> and enjoy :)

Testing:

- Run the command:
`python manage.py test`

Admin Panel:

Feel free to add / edit / delete entries as you wish.

<http://localhost:8000/admin/>

Credentials:

- username: admin
- password: admin

(In case you want to create another super user use the command:

`python manage.py createsuperuser`)

Django administration

Site administration

API		
Candidates	+ Add	✎ Change
Jobs	+ Add	✎ Change
Skills	+ Add	✎ Change