

Backend Developer Assignment

Make sure you read the whole document carefully and follow the guidelines in it.

Context

Create a REST API that takes in multiple paragraphs of text as input, stores each paragraph and the words to paragraph mappings on a PostgreSQL database.

Custom User Model

```
{
  "id": "xxx", // user ID (PK)
  "name": "test", // user name
  "email": "john@website.com" // email ID (unique)
  "dob": "", // date of birth
  "createdAt": "" // user created date
  "modifiedAt": "" // user modified date
}
```

Requirements

Given a word to search for, it lists out the top 10 paragraphs in which the word is present

Few points to consider

- Tokenize the words by splitting at whitespace.
- Convert all words to lowercase.
- Index these words against the paragraph they are from.
- Generate a unique ID for every paragraph that is indexed.
- A paragraph is defined by two newline characters.

Example

Sample Input: Two paragraphs separated by two new lines

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Magna ac placerat vestibulum lectus. Elit duis tristique sollicitudin nibh sit amet commodo. Senectus et netus et malesuada fames. Fermentum iaculis eu non diam phasellus vestibulum lorem sed. Dictumst quisque sagittis purus sit amet volutpat consequat mauris. Aliquam ut porttitor leo a diam sollicitudin tempor. Consectetur a erat nam atlectus urna duis convallis. Sed viverra ipsum nunc aliquet bibendum enim facilisis gravida neque.

Maecenas volutpat blandit aliquam etiam erat velit scelerisque. Lectus sit amet est placerat in egestas erat imperdiet. Ante in nibh mauris cursus mattis. Tellus rutrum tellus pellentesque eu tincidunt. Euismod quis viverra nibh cras pulvinar mattis. Proin nibh nisl condimentum id

venenatis a. Quam elementum pulvinar etiam non quam. Arcu dictum varius dui at consectetur lorem donec. Aliquet porttitor lacus luctus accumsan tortor. Duis ut diam quam nulla porttitor massa id.

Sample search

Input - lorem | Output: Paragraph 1 and 2 are returned

Input - Maecenas | Output: Paragraph 2

Functionality

- User should first login via the api
- All subsequent api requests must be authenticated or else they will fail
- The API should follow typical RESTful API design patterns.
- The data should be saved in the PostgreSQL DB.
- Provide proper README with usage instructions.
- Provide proper Swagger/Postman API documentation.

Tech stack

- Use Django Rest Framework
- Use a Relational DB. PostgreSQL is preferred.
- Use Docker and docker-compose (bonus)

Bonus

- Write clear documentation on how it's designed and how to run the code
- Write good in-code comments.
- Follow Google coding standard for python:
<https://google.github.io/styleguide/pyguide.html>
- Write good commit messages, follow the git workflow:
<http://udacity.github.io/git-styleguide/>

What We Care About

Feel free to use any open-source library as you see fit, but remember that we are evaluating your coding skills and problem-solving skills.

Here's what you should aim for:

- Good use of coding & API design best practices.
- Good testing approach.
- Extensible code.
- Your output should follow this requirement document closely

Submission:

Once completed push the code to a private GitHub repo and share it with us. Share the code to the following email Id: ravi.bhalani@codemonk.in