Senior Python Developer Assignment

The objective is to create a Python service that retrieves data from the StackExchange API and calculates some simple statistics.

Your solution is expected to be delivered within one week. The delivery time will not be taken into consideration as long as it is within the given time frame. Feel free to use any tools/libraries that you feel comfortable with. We appreciate simple and clean over complicated solutions.

We hope that you will enjoy working on this problem, and please do not hesitate to contact us if you need any further clarification. Good luck!

Description

The high-level objective is to deliver an application that runs as a long-running service which provides a REST API with an endpoint that takes as input a date/time range, retrieves data from the StackExchange API, calculates some statistics and reports them back to the user. The service should also cache any request and provide the cached results to any subsequent same request.

The service

You can implement the service in any way you want (docker, python-application, etc.). It is required to provide detailed steps for setting up and running the service. Providing a script with the necessary commands is preferable.

Communication with StackExchange API and statistics

The service will have to communicate with the StackExchange API and perform the following tasks:

- Retrieve the StackOverflow answer data for the given date/time range (Endpoint: https://api.stackexchange.com/docs/answers).
- Retrieve the comment data for this set of answers[1] (Endpoint: https://api.stackexchange.com/docs/comments-on-answers).
- Finally, calculate the following statistics:
  - the total number of accepted answers.
  - the average score for all the accepted answers.
  - the average answer count per question.
  - the comment count for each of the 10 answers with the highest score.

REST API

The REST API that will provide the service can be implemented with any technology you may prefer. The endpoint should be provided at http://localhost:5000/api/v1/stackstats and should take two datetime parameters, "since" and "until", as strings. An example of the expected datetime parameters is: "2023-01-02 11:00:00".

Request example

```
curl --request GET \
       --url http://localhost:5000/api/v1/stackstats?since=2023-01-02 10:00:00&
until=2023-01-02 11:00:00
```

[1] *The specific calls should not be limited by the given time range.*

Response body example

```
{
    "total_accepted_answers": 10,
    "accepted_answers_average_score": 23.8,
    "average_answers_per_question": 1.3,
    "top_ten_answers_comment_count": {
        "38149500": 1,
        "38152507": 7,
        "38147398": 5,
        "38142598": 2,
        "38149856": 0,
        "38143675": 3,
        "38143335": 1,
        "38143566": 0,
        "38143884": 9,
        "38143115": 1
    }
}
```

* Output example values are just an example of the output format, do not take them into consideration as real values.

Caching layer

A caching layer should also be implemented. The purpose of caching is to minimize the response time when executing the same request more than once. You can use any database-like technology to implement it. It is up to you, if you want to take the caching layer a step further, but it is not required.

Additional Instructions

- The supported Python version should be >=3.6.
- Focus on writing clear and well-documented code, instead of spending too much time with error handling and edge cases.
- Following programming best practices such as unit testing, code structure, and documentation will be taken into consideration.
- A Markdown formatted README.md file should be included in the project files with instructions regarding:
    - Installation
    - How to run the service and an example request to the API.
    - How to run the test cases (if any are included)
- Deliverable source files should be included in a folder named **python-assignment**, compressed in one of the following common formats:
    - TAR: encode-python-assignment.tar
    - GZIP: encode-python-assignment.gz
    - ZIP: encode-python-assignment.zip