Developer

Please develop a small REST service for retrieving all possible matches of a prefix in a dictionary. The service will provide 2 endpoints:

1. `GET /dictionary?prefix=PREFIX`

Will retrieve all possible matches of a prefix in a dictionary.

```
Example:
    curl --request GET '<https://127.0.0.1:PORT/dictionary?prefix=AAH'>
    will return:
    [
        "AAH",
        "AAHED",
        "AAHING",
        "AAHS"
    ]
```

2. `GET /statistics`

The service will keep statistics on handling the dictionary endpoint, the stats are collected for a single process/instance.

* The statistics doesn't have to be persistant.
* The endpoint Will retrieve a json object of the statistics of the service instance.
averageRequestHandleTimeMs - The average time in ms it took the service to handle the requests.
requestHandledCount - How many requests did the service handled.
wordCount - How many words are in the dictionary.

```
Examples:
    curl --request GET '<https://127.0.0.1:PORT/statistics'>
    will return:
    {
        "averageRequestHandleTimeMs": 0.123,
        "requestHandledCount": 8,
        "wordCount": 204833
    }
```

3. Update dictionary

```
    POST /update_dictionary
    Will recieve a file with a list of words and will update the
dictionary accordingly.

    * No need to update the dictionary for all processes
    * It's ok if the update will not be persistent (the service will
lose the update after restart).
    * Feel free to choose any file format that will be reasonable and
easy to use.
```

Scope and Concerns

- You are free (and encouraged) to use any web resources,
  boilerplate code and 3rd party libraries you want.
- The amount of time invested should not be more than 2-3 hours.
- The solution must work and could be easily run on any machine under reasonable run times.
- The goal of the assignment is to present a code that will pass a Code Review, the code should be clean and readable.
- Please document the steps to run your code,
  assume we never developed in your chosen language or platform.
- Please document your engineering decisions \ assumptions.
- Please use backend development best practices.
- The autocomplete algorithm implementation should be naive and basic.
- The dictionary will be supplied by us, the word count is 204,833 and that is the scale you should expect.
- The service should be developed in any language you can see fit.
- You can deliver the solution in any format easy for you (zip file \ git repo \ etc).

Good luck, feel free to contact us with any question or concerns you have.