

Twitter Storage SRE Coding Challenge

Description:

Your objective for this challenge will be to produce a tool that will query the 'status' page on 1000 **fictitious** servers and create a report based on data within the status page. Each server has a 'status' endpoint that returns JSON data. The details of that JSON data will be provided below.

Your tool should produce the following reports:

- Human readable report output to stdout.
- Computer parseable report output to a file on local disk.

Task Details:

- Input:
 - List of servers:
 - The file 'servers.txt' will include a list of **fictitious** servers. Your solution should read this file for its list of endpoints.
 - Please implement the code required to call the endpoint (as an HTTP GET request) and retrieve the data.
 - For Example: `http://<server_name>.twitter.com/status`
 - Response from endpoint:
 - The file 'responses.txt' will provide example JSON data that you should reference for this exercise.
 - Please **DO NOT** read the 'responses.txt' file in your code. This is just example data so you can see what could be returned from the endpoints. Each endpoint can be considered to reply with a single entry from the list in 'responses.txt'.
- Output:
 - Your tool should scan each endpoint and produce a report that aggregates:
 - Success rate, ordered by Application, then Version
 - Your tool should write its output to standard out in human readable format.
 - Your tool should also write output to a local file that is easily computer parsable (plan for this to be consumed by a downstream application).

Rules/Expectations:

- Please implement this in the language of your choice, but, we would prefer **no shell-scripting for this exercise**.
- You may use online resources (Google, Stack Overflow, books, etc.). Please write this code on your own (do not reach out to any individuals or groups to contribute to your solution).

- The code should be free of compilation and logic errors, and runnable in Linux command line environment.
- In addition to correctness, other areas that are also important are clean and easy to understand code, speed and effective resource utilisation within the confines of the language used for your solution.
- Your solution should include a README with instructions on how to compile/run your code.

Additional Information:

- We expect this challenge to take between one to three hours.
- Please be prepared to discuss your solution if you are selected for an on-site interview. We may ask you to explain design decisions, fix bugs, and extend your solution to address additional use-cases.