REddit Listener

Jack Henry Coding Exercise

# Overview

Use this space to add background context, objectives, and suggested timelines.  
Add inspiration quote here.

# Summary

|  |
| --- |
| 8/6/2024 |
|  |
| Reddit, much like other social media platforms, provides a way for users to communicate their interests etc. For this exercise, we would like to see you build an application that listens to your choice of subreddits (best to choose one with a good amount of posts). You can use this [link](https://redditcharts.com/) to help identify one that interests you.  We’d like to see the exercise completed in the language you are applying for, and you are free to use any 3rd party libraries you would like.  Your app should consume the posts from your chosen subreddit in near real time and keep track of the following statistics between the time your application starts until it ends:  Posts with most up votes  Users with most posts  Your app should also provide some way to report these values to a user (periodically log to terminal, return from RESTful web service, etc.). If there are other interesting statistics you’d like to collect, that would be great. There is no need to store this data in a database; keeping everything in-memory is fine. That said, you should think about how you would persist data if that was a requirement.  To acquire near real time statistics from Reddit, you will need to continuously request data from Reddit's rest APIs.  Reddit implements rate limiting and provides details regarding rate limit used, rate limit remaining, and rate limit reset period via response headers.  Your application should use these values to control throughput in an even and consistent manner while utilizing a high percentage of the available request rate.  It’s very important that the various application processes do not block each other as Reddit can have a high volume on many of their subreddits.  The app should process posts as concurrently as possible to take advantage of available computing resources. While we are only asking to track a single subreddit, you should be thinking about his you could scale up your app to handle multiple subreddits.  While designing and developing this application, you should keep SOLID principles in mind. Although this is a code challenge, we are looking for patterns that could scale and are loosely coupled to external systems / dependencies. In that same theme, there should be some level of error handling and unit testing. The submission should contain code that you would consider production ready.  When you're finished, please put your project in a repository on either GitHub or Bitbucket and send us a link. Please be sure to provide guidance as to where the Reddit API Token values are located so that the team reviewing the code can replace/configure the value. After review, we may follow-up with an interview session with questions for you about your code and the choices made in design/implementation.  While the coding exercise is intended to be an interesting and fun challenge, we are interested in seeing your best work - aspects that go beyond merely functional code, that demonstrate professionalism and pride in your work.  We look forward to your submission! |
|  |
| **Accessing the Reddit API**  To get the API, register [here](https://www.reddit.com/wiki/api/)  Additional documentation can be found [here](https://www.reddit.com/dev/api/) |
|  |
| Architecture |
|  |
| Future Potential Enhancements (After Phase 2)  * Data Visualization via PowerBI or Tableau * Containerization and orchestration of components * Async Processing via Message Queue * Enhanced Terminal UI via Spectre Console (or similar) |
|  |
| Timeline  * Phase 1 (MVP) is recommended to take between 4 and 6 hours * Phase 2 would require a formal planning session to estimate accurately   + DB and API would probably take 1 iteration   + Each Client would take at least 1 iteration depending on desired functionality |
|  |