Our standard interview process includes a programming exercise. There's no time limit on it, as it's intended to be able to fit around your other responsibilities. Simply keep us updated on progress so that we know it’s active.

**Programming Assignment:**

The [Twitter API provides a stream endpoint](https://developer.twitter.com/en/docs/basics/getting-started) that delivers a roughly 1% random sample of publicly available Tweets in real-time. In this assignment you will build an application that utilizes that endpoint and processes incoming tweets to compute various statistics. We'd like to see this as a .NET Core or .NET Framework project, but otherwise feel free to use any libraries or frameworks you want to accomplish this task.

The Twitter API v2 [sampled stream endpoint](https://developer.twitter.com/en/docs/twitter-api/tweets/sampled-stream/introduction) provides a random sample of approximately 1% of the full tweet stream. Your app should consume this sample stream and keep track of the following:

* Total number of tweets received
* Average tweets per hour/minute/second
* Top [emojis](http://en.wikipedia.org/wiki/Emoji) in tweets\*
* Percent of tweets that contains emojis
* Top hashtags
* Percent of tweets that contain a url
* Percent of tweets that contain a photo url (pic.twitter.com or Instagram)
* Top domains of urls in tweets

\* The [emoji-data](https://github.com/iamcal/emoji-data) project provides a convenient emoji.json file that you can use to determine which emoji unicode characters to look for in the tweet text.

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.

It’s very important that when the application receives a tweet it does not block statistics reporting while performing tweet processing. Twitter regularly sees 5700 tweets/second, so your app may likely receive 57 tweets/second, with higher burst rates. The app should process tweets as concurrently as possible to take advantage of available computing resources. While this system doesn’t need to handle the full tweet stream, you should think about how you could scale up your app to handle such a high volume of tweets.

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 Twitter API Token is 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 Twitter API Key:**  
To obtain a Twitter API Key you will need to create/utilize a twitter account with an email address. During the application process you will also be asked to verify your phone number. You can apply here: [Twitter API Access](https://developer.twitter.com/en/apply-for-access)  
  
Below are screenshots of recommended application inputs. These are recommendations based on our team’s previously approved access. It is your decision if you choose to apply these recommendations. Once the application is complete you should receive a email to validate you application. In most cases, you will receive access immediately after confirming your email address.  
  
