

Candidate Technical Assessment - Scripting

Instructions:

1. Review this document
2. Implement Variation **#2a** of this assignment in **C++**
3. Completed project must build and run under **Linux**, **Windows**, or **Mac OS X.**
4. Complete within **1 week**
5. If you have any questions or need clarification on any requirements, please email sydneypolk@gamestop.com

**Table Of Contents**

1. Overview 2

2. The Assignment 2

2.1. Requirement #1 3

2.2. Optional requirement #2 3

2.3. Optional requirement #3 3

3. Deliverables 4

# Overview

The purpose of this assessment is to allow the candidate to demonstrate their software engineering expertise in domains of interest to Spawn Labs.

This assessment should not take more than a couple of hours to complete.

The candidate is free to ask questions of the designated evaluator at Spawn Labs while working on the assessment.

# The Assignment

The assignment is to build a simple consumer of the Twitter Stream API (https://dev.twitter.com/docs/streaming-apis). You will write a program which connects to the user stream of a user you provide, parse the data as it is sent back for a specified amount of time, and prints out statistics based on data mined from the stream. Optionally, you will store the tweets and provide another program which can retrieve tweets that you have processed. Also, optionally, you will handle errors that can be generated during the process.

You must use either perl, python or ruby.

## Requirement #1

The command-line invocation:

./twitter-stream

will do the following:

* Connect with the Twitter Streams User API, GET user, with a hard-coded user and password.
* Listen to the stream for 2 minutes.
  + Get information for the user and the followers of the user. No replies.
  + No stall warnings.
* Print out two lists, one of users, and one of hash tags, in the following format:

Users:

@aa – 3

@ab – 5

@AtWork – 8

Hash Tags:

#abfab – 17

#barbeque – 23

#Chicago – 120

#zanzibar – 1

Note that the lists should be printed in sorted, non-case sensitive order.

## Optional Requirement #2

Add the following command-line parameters, which will override the defaults if provided:

-u/--user – The twitter user.

-p/--password – The twitter password.

-d/--duration – The time that the sampling will occur. This should be in seconds if a raw number is specified. Optionally, parse “2s” as 2 seconds, “2m” as two minutes, “3h” as 3 hours, “2:13” as two minutes, 13 seconds, etc.

## Optional Requirement #3

Add error handling. There are stall warnings that need to be handled. Handle disconnects and the documented HTTP errors that can be returned.

## Optional Requirement #4

This task is large, open-ended, and completely optional. You can take it as far as you like. Note that SQL knowledge is not a pre-requisite for this position, but is for this task.

Store the tweets in a SQL database. Write another utility that can generate the above output after a sampling, using SQL instead of language built-in data structures. Publish the SQL schema. Write another utility that uses SQL statements that display all tweets by a given user, or with a given hash tag. The command-line parameters, database technology, and output formatting are left to you.

Please provide which database this is written for, and scripts to generate the schema.

# Deliverables

The following should be delivered as a single .ZIP archive:

* Source code in the requested target language(s)
* Design Documentation, i.e., a narrative description of the design including models/diagrams (e.g. UML)
* Build/Run Instructions on the requested target platform. If there are 3rd party dependencies, be sure to call these out.

How the assignment will be evaluated:

* Design Review:
  + Is the design narrative understandable?
  + Is there a consistent style to the models/diagrams?
  + Does the design meet the requirements?
    - If not, does the design call out and recognize its deficiencies
* Implementation Review:
  + Does the implementation match the design?
  + Does the implementation build and execute on the target environment?
  + Does the implementation “work”?
* Code Review:
  + Are code comments understandable?
  + Is there a consistent coding style in the language(s) used?