Platform Engineer Challenge

# Introduction

The Floow uses Java and several other technologies to process data as part of their Telematics Platform. To ensure reliability and performance, many components of the service must operate in a distributed fashion to balance load and responsibility of continuous operation. This test is designed to assess your ability to create a system that achieves these objectives on a smaller scale.

# The Challenge

The functional goal of the challenge is to analyse a file with a large body of text and to count unique words so that the most common, least common and other statistically interesting words can be identified.

The technical goal of the challenge is to create a system that can be distributed, and scale reasonably easily.

You are required to produce a program that counts the words in a file and safely saves the counts to a MongoDB server. Ideally the program will support execution on multiple servers that communicate via common means (eg a Mongo collection) and work together to break down the workload.

The challenge should be capable of being executed as a JAR on a number of servers that talk to one MongoDB instance using the following command on each server:

|  |
| --- |
| java –Xmx8192m -jar challenge.jar –source dump.xml –mongo [hostname]:[port] –id [server hostname/id string] |

As a result, your solution should be provided as an executable JAR which expects the arguments described above, and respects the restrictions implied by them.

For the purposes of assessment we can run pre and/or post processes if required but they must be clearly defined as part of the submission. A means to view the final results (word counts) must also be described or provided as appropriate.

Your solution should be produced using a git repository, and you should utilise git the same way that you would use version control in your current job.

The following bonus goals will be taken into consideration if achieved:

* The ability to create a single program that handles pre and post processing as required.
* The ability to handle a variable number of processes/servers.
* The ability to handle failure and continuation of one or more processes/servers.
* Notes considering trade offs between efficiency and accuracy

# Test Data Set

While you can test and develop using any data set, this challenge will be assessed using the following (large) dataset which can be downloaded from [*https://dumps.wikimedia.org/enwiki/20170701/*](https://dumps.wikimedia.org/enwiki/20170701/)

* *enwiki-20170701-pages-articles-multistream.xml.bz2*

***Please Note:*** *These links change from time-to-time and this one may no longer be the most up-to-date.*

If it is difficult to acquire we can provide a copy of this dataset – Wikipedia dumps can be unstable at times.

You will not be expected to account for the fact that it is XML based, no extra credit will be given for doing so.

# The Deadline & Submission

Your submission must include:

* Complete source code hosted in a git repository such as [GitHub](https://github.com/)[[1]](#footnote-0) or [Bitbucket](https://bitbucket.org/)[[2]](#footnote-1).
* An executable JAR which can be run using the command defined in “The Challenge” section above. You may provide submit your jar as part of the repository or separately (e.g. zipped attachment to your email).
* Additionally you must provide any and all instructions required to successfully run your solution and describe how to view the final results (the word counts).

Your submission will be assessed internally following which you will be invited along to present and discuss the submission and results.

1. GitHub lets you have an unlimited number of free repositories [↑](#footnote-ref-0)
2. Bitbucket lets you have an unlimited number of free and private (if shared with no more than 5 people) repositories. [↑](#footnote-ref-1)