**Process and results:**

* How does your script work?
* This is a java program which internally uses **curl** library (to get API response) and **gnuplot** library (for the plot).
* The program reads certain information from the property file (config.props) and retrieves the data from the server with a specific polling interval (i.e. fixed as 1 second).
* Once the response is retrieved, the program extracts (using json library) the sequence number and time from the response and writes the data into a file (as per value of number of attempts).
* After writing the data to a file, the programs creates a gnuplot script and invokes gnuplot command (passes gnuplot script as s parameter) to show the plot.
* The program also shows the minimum, maximum and average time taken for a new ledger to be validated during the span of time captured.
* How did you decide on your polling interval?
* After analyzing the responses from the server (using a simple batch script which fetched the data continuously in a loop), I observed that approximate time taken for a new ledger to be validated was 2-3 seconds. I decided to use polling interval of 1 second. If I reduce the polling timer, the program tries to connect very frequently and the average (min. and max as well) time increases. Similarly, if I increase the polling timer, average (min. and max as well) time increases due to the sleep interval.
* What do the results tell you?
* It takes around 2-3 seconds average for a ledger to be validated. However, I feel this question should be more specific.
* What might explain the variation in time between new ledgers?
* This variation may be because of number of consensus rounds. Time may increase with number of rounds (and vice versa).

**Bonus question #1**: Enhance your script to calculate the min, max, and average time that it took for a new ledger to be validated during the span of time captured.

* Done (as explained above)

**Bonus question #2**: There are some other (better) ways that you could use the rippled API to find how long each ledger took to close/validate. Using the API documentation, find and describe one of these methods (you don’t need to actually implement it).

* Instead of using the command line interface, I have used JSON-RPC API which I think is a better way.

**How to run the program**

1. This program has been developed and tested on Windows 10.
2. Ensure that the Windows machine is connected to the internet.
3. Ensure that java is installed on Windows machine and path is set.
4. Download and extract the zip file (shared over github link) in a temporary folder.
5. Open the config.props file present in the temporary directory and specify the number of attempts (preferred value is above 5) and gnuplot library path (upto bin with double back-slashes)
6. Open a command prompt and navigate to the temporary folder. Execute the below command:

*java –jar ShowLedgerSeqStats.jar config.props*

**Screenshots for reference**









