**# Backend Engineer Test (Golang)**

This task is designed to help you work with concepts frequently used in backend systems development. We don't expect you to be fluent in language and concepts so take you time to familiarize yourself while attempting this task.

**You will have 4 days to submit this task.**

## Programming Language

[Go v1.17](https://go.dev/)

## Tasks

Create a TCP server that listens on port `9000` and publishes stock ticks. Each tick is a JSON object with the following fields:

```json

{

"time": "2020-01-01T00:00:00.000Z",

"symbol": "AAPL",

"open": 120.00,

"high": 150.00,

"low": 110.00,

"close": 121.00,

"volume": 1300000

}

```

Server will publish ticks for 10 symbols that will be randomly generated. Server will create a cache of 10 stocks will following default values:

```json

{

"time": "2020-01-01T00:00:00.000Z", // current time

"symbol": "XAFG", // randomdly generated symbol

"open": 100.00,

"high": 100.00,

"low": 100.00,

"close": 100.00,

"volume": 10000

}

```

After every 100 milseconds, server will pick a random stock from the cache, update the stock values and publish the tick.

Updating stock values will follow following format:

- Update `time` to current

- Pick a random number between -10% and +10% of `close` price and add it to `close` price

- If new `close` is higher than the `high`, update the `high`

- If new `close` is lower than the `low`, update the `low`

- Pick a random number between 0 to 1000 and add it to the volume

Ticks will be published as newline delimited JSON objects.

## Evaluation Criteria

- Project Completion

- Project Structure

- Code Organization

- Code Quality

## Submission

You are required to create a public repository on \*\*GitHub.com\*\*, place your source code and relevant dataset into that repository and share the public repository URL with us.

## Evaluation

We will evaluate your submission on following basis:

- Task Completion

- Code Organization

- Data Structures

## Resources

[Effective Go](https://golang.org/doc/effective\_go.html)

[Go by Example: JSON](https://gobyexample.com/json)

[Network Programming with Go: A TCP Server with a Custom Protocol](https://www.youtube.com/watch?v=yW1ltZidh7g)

[How to Use Netcat Commands: Examples and Cheat Sheets](https://www.varonis.com/blog/netcat-commands/)