

Coding Task – Remote Key/Value Store

Overview

Produce a program capable of storing and retrieving items of data from a lookup table stored in memory. The lookup table should be located on a remote machine and accessed over a network connection. For simplicity, it can be assumed the data is indexed by a string and that the payload (value) is also a string.

The server program should expose a network service, and the client program should be able to communicate with the server to access data. For example, it should be possible to run the server program on a machine as follows:

```
# Runs the server
$ ./server
```

On the same machine (or a second machine), it should be possible to store items of data and retrieve them using a second “client” program. The interface should look roughly as follows:

```
# Stores a value against a specific key in the server located at "node".
$ ./client <node> set <key> <value>
```

```
# Retrieves a value for a specific key and prints it, if found.
$ ./client <node> get <key>
<value>
```

Below is an example of what should be expected to work. In this example, the server is being used to store the names of capital cities for a country.

```
$ ./client 127.0.0.1 set norway oslo
$ ./client 127.0.0.1 set denmark copenhagen
$ ./client 127.0.0.1 get norway
oslo
$ ./client 127.0.0.1 get sweden
error: not found
```

Submission

Submit the following files:

- A single *.zip* or *.tar.gz* file with the source code for your “server” and “client” program.
- A short *README* file describing how to build (if necessary) and run the code.

Guidelines

The task will be used as the basis for a technical interview to evaluate general understanding of software development. This task is purposefully very vague about implementation details. Use it to demonstrate your ability to approach a problem and construct the code to solve the problem, in the manner you are most confident.

- Use any operating system and development tool you like.
- Use any programming language you like e.g. Python, C++, C, Go, Rust.
- Use any libraries you need to complete the task quickly.

However, avoid simply re-use an existing project which implements a server in this fashion (e.g. an existing key/value database). Code is expected for the server/client programs.

- Use any network protocol and associated library.

Choose a tool that you are most familiar with or that makes the task most straightforward (HTTP, gRPC, TCP, etc). We are not so interested in the implementation of a network protocol, just use of a suitable tool to implement the client/server requests.

- **Extra credit:** If time, implement an API of your choice between Client and Server
- **Extra credit:** If time, package the server and run it in a Docker container