Programming assignment 3

Getting started

All your submissions should be implemented in Go unless the problem specifies something different. You can download Go at https://go.dev/dl/. You can also use various package managers to install it, e.g., HomeBrew on macOS. Note that you will need Go 1.22 to run the code provided for Problem 1.

Problems

Problem 1

Download `srv.go` and `test.py` from Moodle. Your task is to create a client (in Go) for the provided server. The server is a key-value store where you can store keys (`put`), get the value stored for a key (`key`), and delete a key (`delete`). The operations return reasonable values: 200 for success, 201 for a created key, and 404 for errors.

Your client should support functions for the three operations and a main program that shows that the client works. Use the Python test program as inspiration.

You are allowed to change the server, e.g., the port it runs on or add functionality. Note that the client does not support concurrent (write)access to the keys (put). This is trivial to fix with, e.g., an RWMutex.

Problem 2

Use the server from Problem 1 as inspiration and create a server that stores objects of some kind (you decide, e.g., students or bookings). Your server must support CRUD (Create, Read, Update, and Delete). Use reasonable endpoints (e.g. `/create`). Use JSON to represent your data objects.

All data should be stored in a database. Think about how you handle concurrency in your requests and database accesses. Describe your application and its design in the report. Your design and implementation, including the scope of your application, will determine your final grade. You do not need to submit any database, but the design should be described in the report.

Inlämning

Submit your solutions as a single zip file via Moodle no later than 17:00 on March 22, 2024 (cutoff 08:00 March 25). This is a group assignment that can be done in groups of one or two students. Your submission should contain well-structured and organized Go code for the problems with a README.txt (or .md) file describing how to compile and run the Go programs and a PDF report describing your design from Problem 2.