

Assignment 3 report

Jesper Wingren

Jw223rn@student.lnu.se

1dt909

Innehållsförteckning

1.	Server and client for hotel booking	3
1.1	<i>Setup</i>	3

1. Server and client for hotel booking

1.1 Setup

The main program is simple, 1 to create a booking, 2 to update a booking, 3 to delete a booking and 4 to get a booking. I have a booking json object as data structure with a MySQL server. Here is my SQL table:

	id	cust_name	cust_phone	datefrom	dateto	room_nr
►	32	Jesper Wingren	0101000	2024-03-21	2024-03-25	105

I have a server written in go which talks to the database. I first establish my connection to the database using constants. After that i setup my http using the http package in go. I then set 4 different methods of requesting, PUT, GET, DELETE and UPDATE. These are built up in calling the correct methods that handles the request. It takes a responsewriter and request as arguments and by using these we can do the correct steps. When getting the value key is used to for example send the id with the http request. This can easily be found using pathvalue from the rquest wh then execute the correct database query and writes to the responsewriter the status code, either statusOK, StatusCreated or if its a fail send the correct error. This makes debugging later alot easier. In the main we then call listenAndServe which waits for requests to come into the server using a given port. When i then close the program i also close the database using defer.

My program2 file or the client as in previous assignment is built to create these requests which are sent to the server. Its a bit different for get while the others are similiar. The get method uses the build in http.Get method to create a http request while the others have to specify what typ of request top be created. This request is then sent to the server using Do which sends the request to the server. Here i also have implemented error checks for both creating the request and sending the request. Before returning anything i check that the statuscode is what its supposed to be before returning. If the statuscode is incorrect it returns false which gives an error message otherwise everything worked.

To handle concurrency i simply put in locks in the delete and update methods which are the only ones that could concurrently create problems. The sql import also handles concurrency correctly via the built in function. Using locks could create problems if the dataload sent to the database is huge but in this case there is no problem because the load sent to the database is small. This could cause problems with scalability.

