Continuous Delivery

Assignment 02

Harald Pinsker (S1910455003)

1. Go REST API

Task: Follow this tutorial <https://semaphoreci.com/community/tutorials/building-and-testing-a-rest-api-in-go-with-gorilla-mux-and-postgresql> and document the progress.

* 1. Setup

I installed Go and Postgres (incl. the required environment variables) and started the Postgres database server.

* 1. Git

I created a new repo on GitHub: <https://github.com/HPin/go-rest-api>

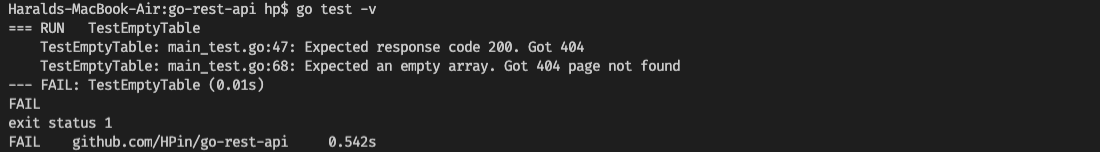
Then I cloned it and initialized the Go modules as described in the tutorial with

go mod init

* 1. Write the code
     1. Test 1: Without any test cases



* + 1. Add test case for empty tables
    2. Test 2: TestEmptyTable



* + 1. Add test cases for all other api operations
    2. Test 3: Test all api operations

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

* + 1. Add the code with the actual app functionality
    2. Create routes and route handlers

Note: the tuturial code had one flaw in the Run function of the App struct where the port parameter was not used and the port was hardcoded instead, so I changed this:

func (a \*App) Run(addr string) {

log.Fatal(http.ListenAndServe(":8010", a.Router))

}

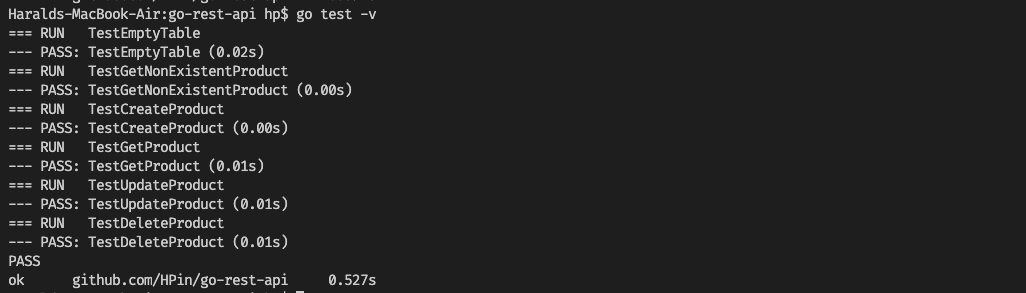
to this:

func (a \*App) Run(addr string) {

log.Fatal(http.ListenAndServe(addr, a.Router))

}

* + 1. Test 4: Test of the fully functioning app



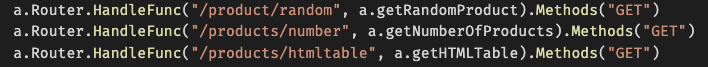
All tests passed.

* 1. Add Functionality

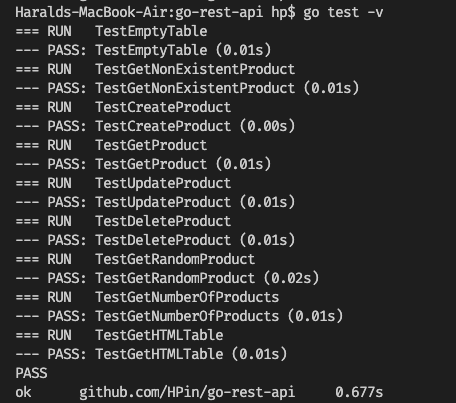
Task: Add 2-3 small features

I implemented three different new functions/endpoints:

1. Get a random product
2. Get the number of products in the database
3. Get the products in a simple HTML table (<table><tr><th></th></tr></table>)



Test again:



All tests passed.