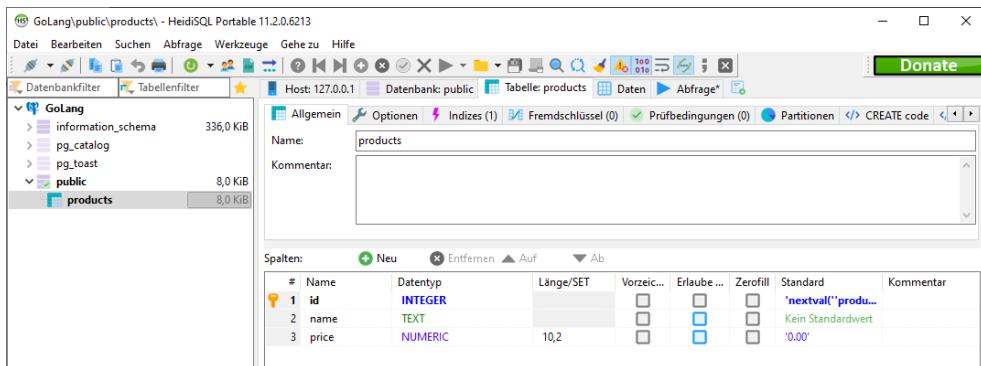


Exercise 02 - Programming in Go

<https://github.com/mleister97/cicd-golang>

Setup

- Installed Go <https://golang.org/doc/install>
- Installed Postgres <https://www.postgresql.org/download>
- Heidi SQL <https://www.heidisql.com/download.php>

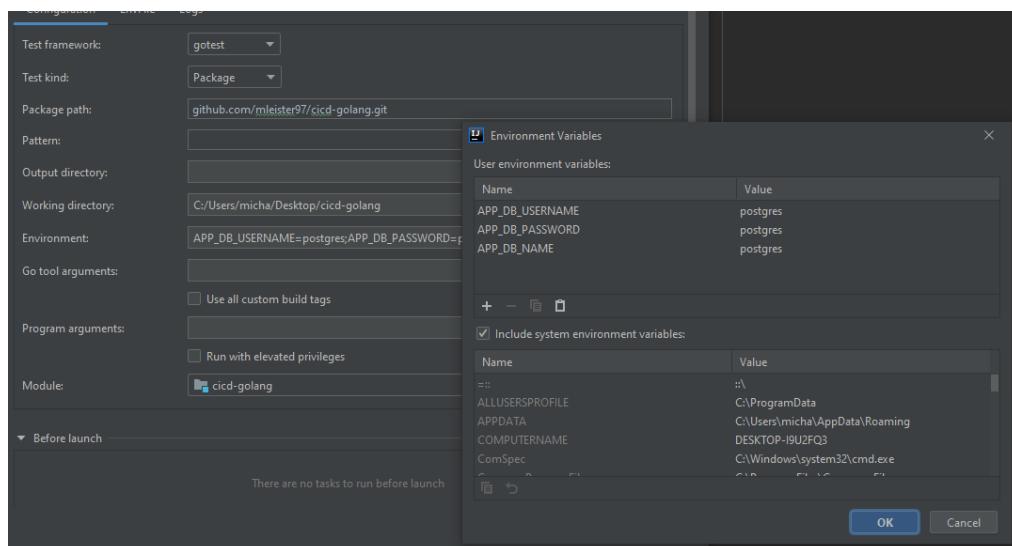


```
go mod init github.com/mleister97/cicd-golang
go get -u github.com/gorilla/mux
go get -u github.com/lib/pq
```

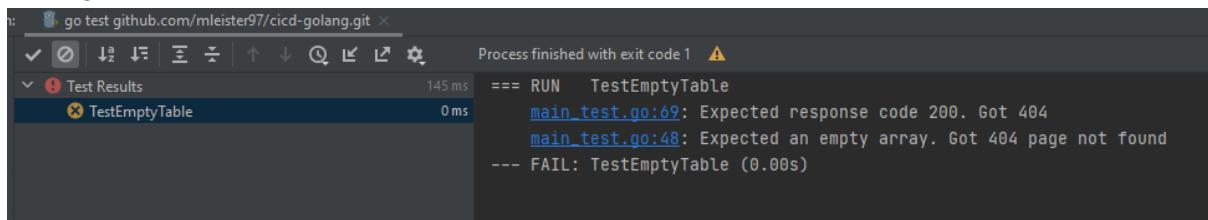
I was facing the following issue:

2021/03/25 14:52:56 pq: SSL is not enabled on the server

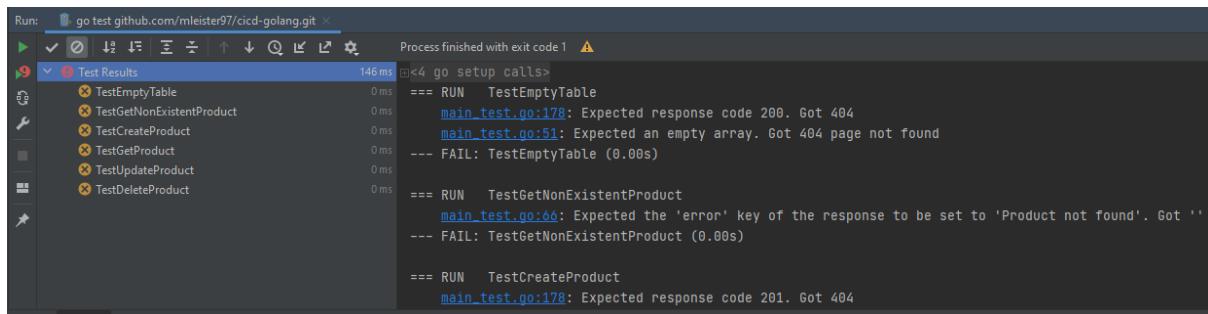
Issue was: Environment variables were not set.



Testing:



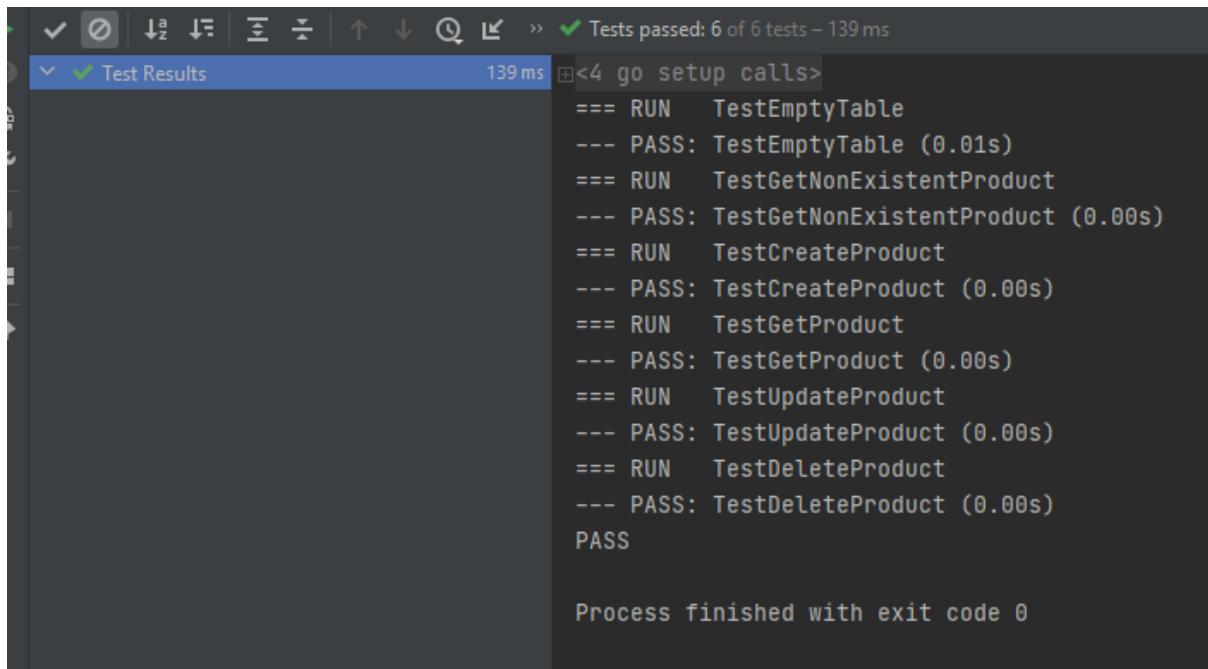
```
Process finished with exit code 1
    === RUN   TestEmptyTable
    0ms      main_test.go:69: Expected response code 200. Got 404
    main_test.go:48: Expected an empty array. Got 404 page not found
--- FAIL: TestEmptyTable (0.00s)
```



```
Process finished with exit code 1
    === RUN   <4 go setup calls>
    0ms      main_test.go:178: Expected response code 200. Got 404
    main_test.go:51: Expected an empty array. Got 404 page not found
--- FAIL: TestEmptyTable (0.00s)

    === RUN   TestGetNonExistentProduct
    0ms      main_test.go:66: Expected the 'error' key of the response to be set to 'Product not found'. Got ''
--- FAIL: TestGetNonExistentProduct (0.00s)

    === RUN   TestCreateProduct
    0ms      main_test.go:178: Expected response code 201. Got 404
```



```
Tests passed: 6 of 6 tests - 139 ms
    ===<4 go setup calls>
    === RUN   TestEmptyTable
    --- PASS: TestEmptyTable (0.01s)
    === RUN   TestGetNonExistentProduct
    --- PASS: TestGetNonExistentProduct (0.00s)
    === RUN   TestCreateProduct
    --- PASS: TestCreateProduct (0.00s)
    === RUN   TestGetProduct
    --- PASS: TestGetProduct (0.00s)
    === RUN   TestUpdateProduct
    --- PASS: TestUpdateProduct (0.00s)
    === RUN   TestDeleteProduct
    --- PASS: TestDeleteProduct (0.00s)
    PASS

    Process finished with exit code 0
```

Custom Feature #1

Add a “available” attribute to the database. Modify the existing code. Only return products which are available in the GET routes. Write test-cases for the availability.

Changes can be found in a separate commit called “*Add availability feature*”.

Custom Feature #2

I will add another endpoint function that will allow me to ONLY update the price. If you pass more details to the request, they will be ignored. Write test-cases for the new implemented route. Changes can be found in a separate commit called “*Add update price only feature*”.