Documentation Microservice Reset

Goal: Allow to us to reset the Boolean in our database MySQL (set the value to 0).

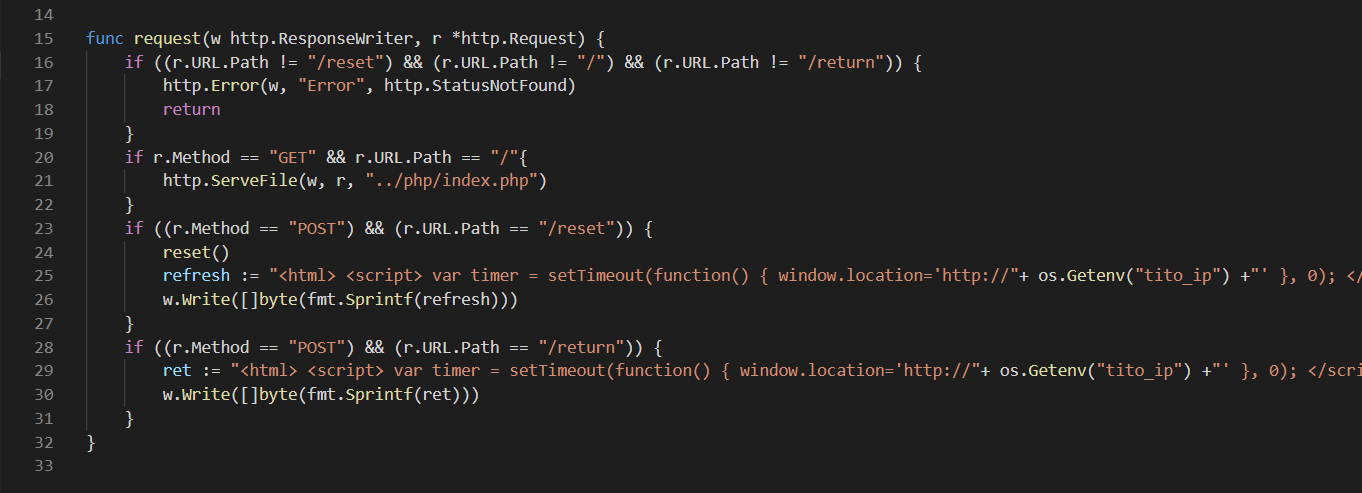
First we go with the main function in the main.go file:



Here we declare our URL that we will work on and the port that we will use: 8080.

We call the function request which will manage our http request.

Method.go file.



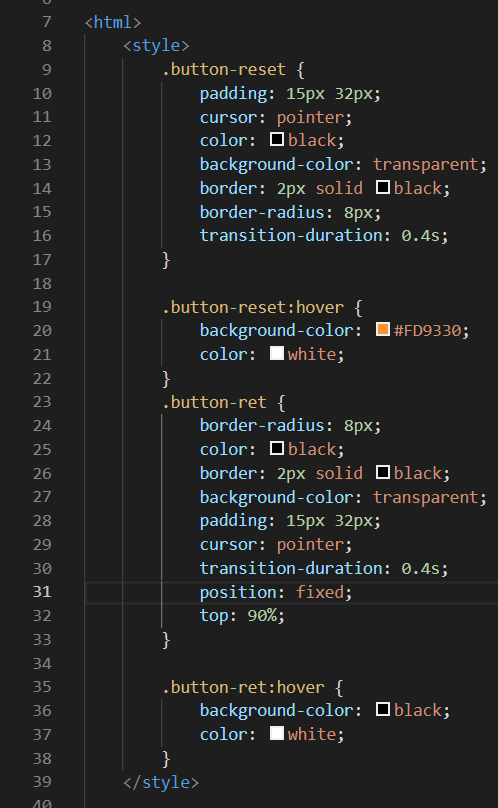
Line 25 : refresh := "<html> <script> var timer = setTimeout(function() { window.location='http://"+ os.Getenv("tito\_ip") +"' }, 0); </script> </html>"

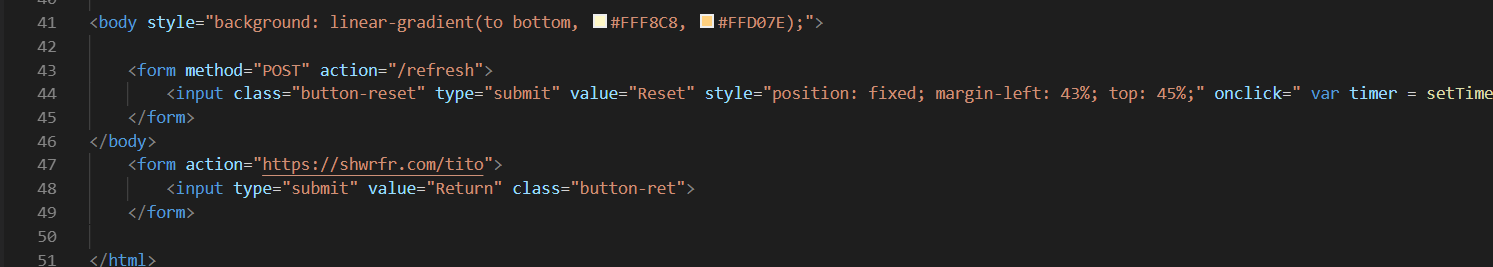
Line 29: ret := "<html> <script> var timer = setTimeout(function() { window.location='http://"+ os.Getenv("tito\_ip") +"' }, 0); </script> </html>"

We are using env variable set in the Dockerfile to redirect the URL.

First, we check if the URL is correct. Then, we check which http request we get.

If the http request is GET and the URL is “/”, we display index.html file.

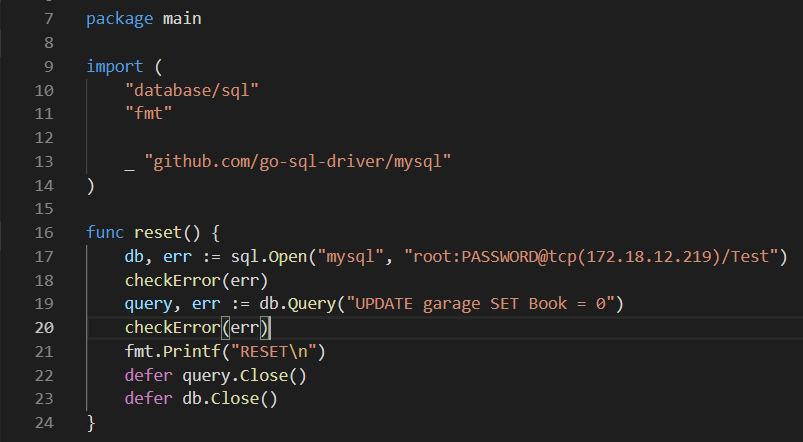




Line 43: <input class="button-reset" type="submit" value="Reset" style="position: fixed; margin-left: 43%; top: 45%;" onclick=" var timer = setTimeout(function() { window.location='https://shwrfr.com/tito/' }, 2000);">

If our http request is POST and the URL is “/reset”, we launch a function name reset.

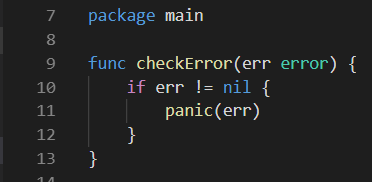
Reset.go file:



This function allow to us to connect to MySQL database and update the boolean.

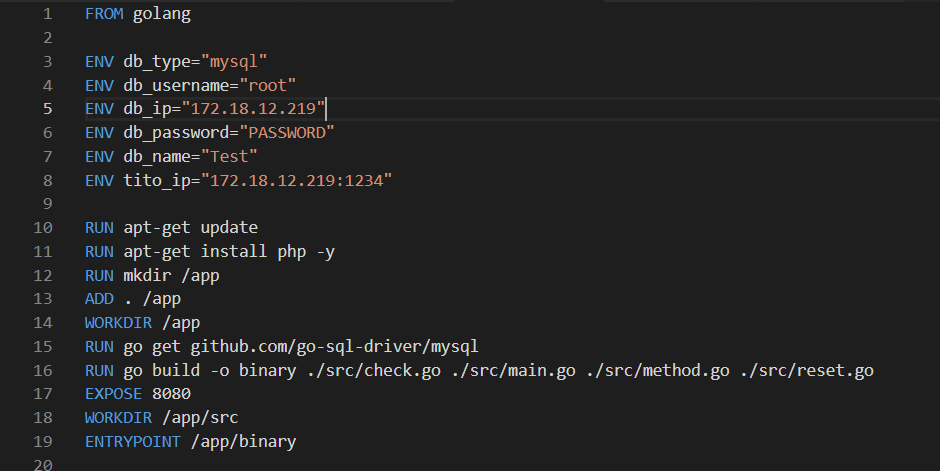
We also check if the is an error when we launch our MySQL request with the checkError function:

Check.go file:



And finally, if we have a POST request with a URL == “/return”, we return to the Tito main page. 😊

Dockerfile:



This Microservice allow to us to reset our Boolean. With this microservice, when all the cars are book, we can reset them to 0 and display them with the booking microservice.

Ip: 172.18.12.219:5555.