



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

## Лабораторная работа № 4 по курсу «Компьютерные сети» «Proxy-server на go»

Студент группы ИУ9-32Б Тараканов В. Д..

Преподаватель Посевин Д. П.

Москва 2024

# 1 Задание

Нужно разработать проху-server на языке go для отображения сайтов

## 2 Результаты

Исходный код программы представлен в листингах 1– ??.

Листинг 1: main.go

```
1 package main
2
3 import (
4     "bytes"
5     "fmt"
6     "github.com/hashicorp/go-getter/v2"
7     "golang.org/x/net/context"
8     "golang.org/x/net/html"
9     "net/http"
10    "strings"
11 )
12
13 var ip = "192.168.31.116:9457"
14 var global_ip = "http://46.138.248.69:9457"
15
16 func replace(doc *html.Node, baseURL string) error {
17     var crawler func(*html.Node)
18     crawler = func(n *html.Node) {
19         if n.Type == html.ElementNode {
20             for i, attr := range n.Attr {
21                 if attr.Key == "href" || attr.Key == "src" || attr.Key == "
action" {
22                     if attr.Val != "/" {
23                         if !strings.HasPrefix(attr.Val, "http") {
24                             fmt.Println("Changed", n.Attr[i].Val, "to", global_ip
+ "/" + baseURL + attr.Val)
25                             n.Attr[i].Val = global_ip + "/" + baseURL + attr.Val
26                         } else {
27                             fmt.Println("Changed", n.Attr[i].Val, "to", global_ip
+ "/" + attr.Val)
28                             n.Attr[i].Val = global_ip + "/" + attr.Val
29                         }
30                     } else {
31                         if !strings.HasPrefix(attr.Val, "http") {
```

```

32         fmt.Println("Changed", n.Attr[i].Val, "to", global_ip
+ "/" + baseURL)
33         n.Attr[i].Val = global_ip + "/" + baseURL + attr.Val
34     } else {
35         fmt.Println("Changed", n.Attr[i].Val, "to", global_ip
+ "/" + n.Attr[i].Val)
36         n.Attr[i].Val = global_ip + "/" + attr.Val
37     }
38 }
39
40 }
41 }
42 }
43 for c := n.FirstChild; c != nil; c = c.NextSibling {
44     crawler(c)
45 }
46 }
47 crawler(doc)
48 return nil
49
50 }
51
52 func handler(w http.ResponseWriter, r *http.Request) {
53     path := r.URL.Path
54     fmt.Println(path)
55     path = strings.TrimPrefix(path, "/")
56     path = strings.Replace(path, "/", "//", 1)
57     isHtml := false
58     if strings.HasSuffix(path, ".html") || (!strings.HasSuffix(path,
".css") &&
59     !strings.HasSuffix(path, ".js") && !strings.HasSuffix(path, ".png
") &&
60     !strings.HasSuffix(path, ".gif") && !strings.HasSuffix(path, ".jpg
") && !strings.HasSuffix(path, ".jpeg") &&
61     !strings.HasSuffix(path, ".ico") && !strings.HasSuffix(path, ".svg
") && !strings.HasSuffix(path, ".ttf") && !strings.HasSuffix(path,
".woff") && !strings.HasSuffix(path, ".woff2")) {
62
63         isHtml = true
64     }
65     if isHtml {
66         get, err := http.Get(path)
67         if err != nil {
68             fmt.Println(err)
69             return
70         }

```

```

71     doc, err := html.Parse(get.Body)
72     fmt.Println("Got page")
73
74     baseURL := path[:strings.LastIndex(path, "/")+1]
75     err = replace(doc, baseURL)
76     fmt.Println("Changed links")
77     if err != nil {
78         fmt.Println(err)
79         return
80     }
81     var buf bytes.Buffer
82     if err := html.Render(&buf, doc); err != nil {
83         http.Error(w, "Error when generating HTML: "+err.Error(), http
84             .StatusInternalServerError)
85         return
86     }
87     w.Write(buf.Bytes())
88 } else {
89     fmt.Println("Picture")
90     filename := strings.Split(path, "/")
91     _, err := getter.GetFile(context.Background(), "./static/"+
92 filename[len(filename)-1], path)
93     if err != nil {
94         fmt.Println(err)
95     } else {
96         http.ServeFile(w, r, "./static/"+filename[len(filename)-1])
97     }
98 }
99
100 func main() {
101
102     http.HandleFunc("/", handler)
103
104     fmt.Println("Proxy-server started on port 9457")
105     err := http.ListenAndServe(ip, nil)
106     if err != nil {
107         fmt.Println("Error when starting server:", err)
108         return
109     }
110 }
111

```