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 № 2

по курсу «Компьютерные сети»

«Клиент и сервер HTTP»

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

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

В ходе выполнения лабораторной работы нужно разработать на языке Go HTTP-сервер, который формирует динамические HTML-страницы на основе данных, получаемых с указанного web-сервера.

2 Результаты

Исходный код программы представлен в листингах 1– **??**.

Листинг 1: download.go

```
1  package main
2
3  import (
4  "net/http"
5  "strconv"
6
7  log "github.com/mgutz/logxi/v1"
8  "golang.org/x/net/html"
9  )
10
11 func getAttr(node *html.Node, key string) string {
12   for _, attr := range node.Attr {
13     if attr.Key == key {
14       return attr.Val
15     }
16   }
17   return ""
18 }
19
20 func getChildren(node *html.Node) []*html.Node {
21   var children []*html.Node
22   for c := node.FirstChild; c != nil; c = c.NextSibling {
23     children = append(children, c)
24   }
25   return children
26 }
27
28 func isElem(node *html.Node, tag string) bool {
29   return node != nil && node.Type == html.ElementNode && node.Data
   == tag
30 }
31
32 func isText(node *html.Node) bool {
```

```
33        return node != nil && node.Type == html.TextNode
34      }
35
36      func isDiv(node *html.Node, class string) bool {
37        return isElem(node, "div") && getAttr(node, "class") == class
38      }
39
40      type Item struct {
41        Ref, Title string
42      }
43
44      func readItem(item *html.Node) *Item {
45        if a := item.FirstChild; isElem(a, "a") {
46          return &Item{
47            Ref:   getAttr(a, "href"),
48            Title: getAttr(a, "aria-label"),
49          }
50        }
51        return nil
52      }
53
54      func search(node *html.Node) []*Item {
55        log.Info("begin of parsing")
56        if isElem(node, "div") && getAttr(node, "id") == "app" {
57          log.Info("found app")
58          var items []*Item
59          for c := node.FirstChild; c != nil; c = c.NextSibling {
60            if isDiv(c, "dTSkA6xB commercial-branding") {
61              log.Info("found dTSkA6xB commercial-branding")
62              for d := c.FirstChild; d != nil; d = d.NextSibling {
63                if isDiv(d, "AuRBdDZg") {
64                  log.Info("found AuRBdDZg")
65                  for k := d.FirstChild; k != nil; k = k.NextSibling {
66                    if isElem(k, "section") {
67                      log.Info("found section")
68                      for e := k.FirstChild; e != nil; e = e.NextSibling {
69                        if isDiv(e, "cGZPyk4_") {
70                          log.Info("found cGZPyk4_")
71                          for g := e.FirstChild; g != nil; g = g.
    NextSibling {
72                            if isDiv(g, "zT5wwAPN fQtJ19Ei") {
73                              log.Info("found zT5wwAPN fQtJ19Ei")
74                              for y := g.FirstChild; y != nil; y = y.
    NextSibling {
75                                if isDiv(y, "XSvLK2D0 abGoxuyb") {
76                                  log.Info("found XSvLK2D0 abGoxuyb")
```

```
77                                  items = append(items, readItem(y))
78                                }
79
80                              }
81                              log.Info("total: " + strconv.Itoa(len(items)) + " news")
82                              return items
83                            }
84                          }
85                        }
86                      }
87                    }
88                  }
89                }
90              }
91
92            }
93          }
94        }
95        for c := node.FirstChild; c != nil; c = c.NextSibling {
96          if items := search(c); items != nil {
97            return items
98          }
99        }
100       return nil
101     }
102
103     func downloadNews() []*Item {
104       log.Info("sending request to https://news.rambler.ru/latest/")
105       if response, err := http.Get("https://news.rambler.ru/latest/"); err != nil {
106         log.Error("request to https://news.rambler.ru/latest/ failed", "error", err)
107       } else {
108         defer response.Body.Close()
109         status := response.StatusCode
110         log.Info("got response from https://news.rambler.ru/latest/", "status", status)
111         if status == http.StatusOK {
112           if doc, err := html.Parse(response.Body); err != nil {
113             log.Error("invalid HTML from https://news.rambler.ru/latest/", "error", err)
114           } else {
115             log.Info("HTML from https://news.rambler.ru/latest/ parsed successfully")
116             return search(doc)
```

```
117             }
118           }
119         }
120         return nil
121       }
122
123
```

Листинг 2: server.go

```
1   package main
2
3   import (
4   "html/template"
5   "net/http"
6
7   log "github.com/mgutz/logxi/v1"
8   )
9
10  const INDEX_HTML = `
11  <!doctype html>
12  <html lang="ru">
13  <head>
14  <meta charset="utf-8">
15  <title>list of news from https://news.rambler.ru/latest/</title>
16  </head>
17  <body>
18  {{if .}}
19  {{range .}}
20  <a href="https://news.rambler.ru/latest/{{.Ref}}">{{.Title}}</a>
21  <br/>
22  {{end}}
23  {{else}}
24  Failed to load news
25  {{end}}
26  </body>
27  </html>
28  `
29
30  var indexHtml = template.Must(template.New("index").Parse(INDEX_HTML))
31
32  func serveClient(response http.ResponseWriter, request *http.Request)
     {
33    path := request.URL.Path
34    log.Info("got request", "Method", request.Method, "Path", path)
35    if path != "/" && path != "/index.html" {
36      log.Error("invalid path", "Path", path)
```

```
37          response.WriteHeader(http.StatusNotFound)
38      } else if err := indexHtml.Execute(response, downloadNews()); err !=
        nil {
39          log.Error("HTML creation failed", "error", err)
40      } else {
41          log.Info("response sent to client successfully")
42      }
43  }
44
45  func main() {
46      http.HandleFunc("/", serveClient)
47      log.Info("starting listener")
48      log.Error("listener failed", "error", http.ListenAndServe
       ("185.104.251.226:9870", nil))
49  }
50
51
```

Результат запуска представлен на рисунке 1– 2.

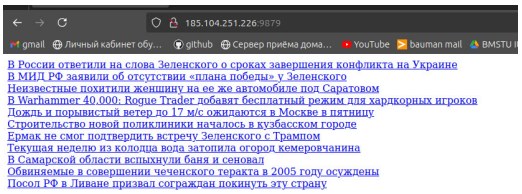

Рис. 1 — Результат

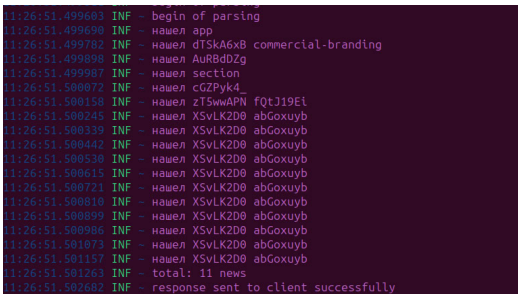

Рис. 2 — Результат