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 Волохов А. В.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Целью данной работы является создание HTTP-клиента и HTTP-сервера на языке Go. Формирование списка последних новостей с https://www.baltimoresun.com/latest/ Исходный код программы представлен в листингах 1– 2– 3.

Листинг 1 — download.go

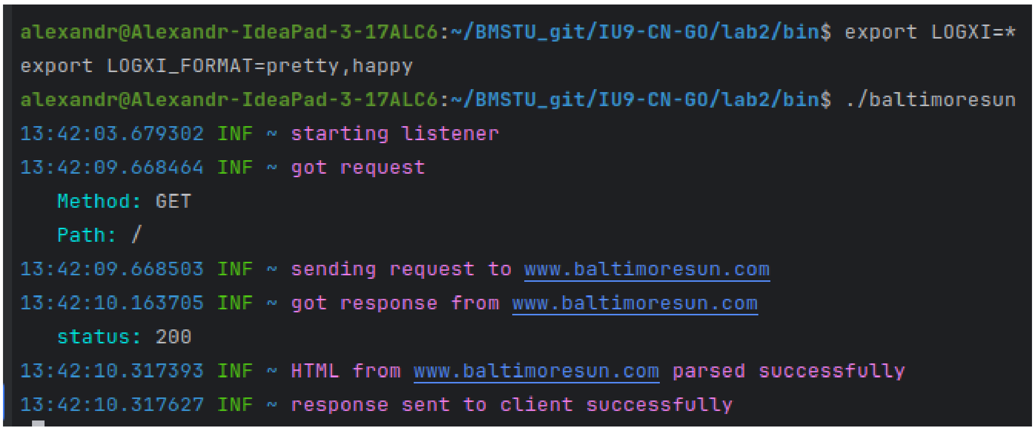
```go
package main

import (
    "github.com/mgutz/logxi/v1"
    "golang.org/x/net/html"
    "io"
    "net/http"
)

func getAttr(node *html.Node, key string) string {
    for _, attr := range node.Attr {
        if attr.Key == key {
            return attr.Val
        }
    }
    return ""
}

func isElem(node *html.Node, tag string) bool {
    return node != nil && node.Type == html.ElementNode && node.Data == tag
}

func isDiv(node *html.Node, class string) bool {
    return isElem(node, "div") && getAttr(node, "class") == class
}

type Item struct {
    Time, Title, SourceLink string
}

func search(node *html.Node) []*Item {
    var items []*Item

    if isElem(node, "article") && (getAttr(node, "class") == "container-fluid row flex_row " ||
        getAttr(node, "class") == "container-fluid row flex_row padding-sm-bottom") {
        item := &Item{}

        for a := node.FirstChild; a != nil; a = a.NextSibling {
            if isElem(a, "a") {
                item.SourceLink = "https://www.baltimoresun.com" + getAttr(a, "href")
                for c := a.FirstChild; c != nil; c = c.NextSibling {
                    if isDiv(c, "headline-wrapper flex flex_col col-sm-xl-7") {
                        for cChild := c.FirstChild; cChild != nil; cChild = cChild.NextSibling {
                            if isElem(cChild, "div") &&
                                (getAttr(cChild, "class") == "promo-headline font_20_custom font_mobile_custom "+
                                    "font_normal story-headline-link color_primary decoration_none ") ||
                                getAttr(cChild, "class") == "promo-headline font_18_custom font_mobile_custom "+
                                    "font_normal story-headline-link color_primary decoration_none " {
                                for tNode := cChild.FirstChild; tNode != nil; tNode =
```

Листинг 2 — download.go - продолжение

```go
tNode.NextSibling {
                  if isElem(tNode, "h2") || isElem(tNode, "h1") {
                    item.Title = tNode.FirstChild.Data
                  }
                }
              } else if isElem(cChild, "div") && getAttr(cChild, "class") == "isDisabled" {
                for cChild2 := cChild.FirstChild; cChild2 != nil; cChild2 = cChild2.NextSibling {
                  if isElem(cChild2, "div") && getAttr(cChild2, "class") == "padding-xs-top" {
                    for tNode := cChild2.FirstChild; tNode != nil; tNode = tNode.NextSibling {
                      if isElem(tNode, "time") {
                        item.Time = getAttr(tNode, "datetime")
                      }
                    }
                  }
                }}}}}}}
    items = append(items, item)
    log.Info("items:", "items", item)
  }

  for c := node.FirstChild; c != nil; c = c.NextSibling {
    childItems := search(c)
    if childItems != nil {
      items = append(items, childItems...)
    }
  }
  return items
}

func downloadNews() []*Item {
  log.Info("sending request to www.baltimoresun.com")
  if response, err := http.Get("https://www.baltimoresun.com/latest/"); err != nil {
    log.Error("request to baltimoresun.com failed", "error", err)
  } else {
    defer func(Body io.ReadCloser) {
      err := Body.Close()
      if err != nil {

      }
    }(response.Body)
    status := response.StatusCode
    log.Info("got response from www.baltimoresun.com", "status", status)
    if status == http.StatusOK {
      if doc, err := html.Parse(response.Body); err != nil {
        log.Error("invalid HTML from www.baltimoresun.com", "error", err)
      } else {
        log.Info("HTML from www.baltimoresun.com parsed successfully")
        return search(doc)
      }
    }
  }
  return nil
}
```

Листинг 3 — server.go

```
package main

import (
  "github.com/mgutz/logxi/v1"
  "html/template"
  "net/http"
)

const INDEX_HTML = `
    <!doctype html>
    <html lang="ru">
        <head>
            <meta charset="utf-8">
            <title>                                    www.baltimoresun.com</title>
        </head>
        <body>
            {{if .}}
                {{range .}}
                    {{.Time}}
                    <a href="{{.SourceLink}}">{{.Title}}</a>
                    <br/>
                {{end}}
            {{else}}
                                                                            !
            {{end}}
        </body>
    </html>
    `

var indexHtml = template.Must(template.New("index").Parse(INDEX_HTML))

func serveClient(response http.ResponseWriter, request *http.Request) {
  path := request.URL.Path
  log.Info("got request", "Method", request.Method, "Path", path)
  if path != "/" && path != "/index.html" {
    log.Error("invalid path", "Path", path)
    response.WriteHeader(http.StatusNotFound)
  } else if err := indexHtml.Execute(response, downloadNews()); err != nil {
    log.Error("HTML creation failed", "error", err)
  } else {
    log.Info("response sent to client successfully")
  }
}

func main() {
  http.HandleFunc("/", serveClient)
  log.Info("starting listener")
  log.Error("listener failed", "error", http.ListenAndServe("127.0.0.1:6060", nil))
}
```



```
alexandr@Alexandr-IdeaPad-3-17ALC6:~/BMSTU_git/IU9-CN-GO/lab2/bin$ export LOGXI=*
export LOGXI_FORMAT=pretty,happy
alexandr@Alexandr-IdeaPad-3-17ALC6:~/BMSTU_git/IU9-CN-GO/lab2/bin$ ./baltimoresun
13:42:03.679302 INF ~ starting listener
13:42:09.668464 INF ~ got request
   Method: GET
   Path: /
13:42:09.668503 INF ~ sending request to www.baltimoresun.com
13:42:10.163705 INF ~ got response from www.baltimoresun.com
   status: 200
13:42:10.317393 INF ~ HTML from www.baltimoresun.com parsed successfully
13:42:10.317627 INF ~ response sent to client successfully
```

Рис. 1 — Пример работы сервера