



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 № 0
по курсу «Компьютерные сети»
«Разработка простейшего web-сервера»

Студент группы ИУ9-32Б Лавров Р. Д.

Преподаватель Посевин Д. П.

Moskva 2025

1 Задание

Рассматривается задача разработки web-сервера на языке GO на основе пакета net/http.

2 Результаты

Исходный код простейший программы приведенный на лекциях ??.

Листинг 1 — Пример реализации класса

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7
8     "github.com/SlyMarbo/rss"
9 )
10
11 func HomeRouterHandler(w http.ResponseWriter, req *http.Request) {
12     q := req.URL.Query()
13     data := q.Get("data")
14     fmt.Println(data)
15
16     feed, err := rss.Fetch("https://tvsamara.ru/rss/")
17     if err != nil {
18         log.Fatalf("Error fetching RSS: %v", err)
19     }
20
21     html := "<html><body>"
22     html += fmt.Sprintf("<h1>%s</h1>", feed.Title)
23     html += fmt.Sprintf("<p>%s</p>", feed.Description)
24     html += fmt.Sprintf("<p>Количество новостей: %d</p>", len(feed.Items))
25     html += "<hr>"
26
27     for i, item := range feed.Items {
28         html += fmt.Sprintf("<div style='margin-bottom: 20px; padding: 10px; border: 1px solid #ccc;'>")
29         html += fmt.Sprintf("<h3>Новость %d</h3>", i+1)
30         html += fmt.Sprintf("<p><strong>Заголовок:</strong> %s</p>", item.Title)
31         html += fmt.Sprintf("<p><strong>Ссылка:</strong> <a href='%s'>%s</a></p>", item.Link, item.Link)
32         html += "</div>"
33     }
34
35     html += "<br><a href='http://localhost:3000/another_page'>Перейти на другую страницу</a>"
36     html += "</body></html>"
```

Листинг 2 — Пример реализации класса

```
1 w.Header().Set("Content-Type", "text/html; charset=utf-8")
2 w.Write([]byte(html))
3 }
4
5 func AnotherRouterHandler(w http.ResponseWriter, req *http.Request) {
6     html := `<html><body>
7         <h1>Другая страница</h1>
8         <a href='http://localhost:3000/'>Вернуться</a>
9         </body></html>`
10
11    w.Header().Set("Content-Type", "text/html; charset=utf-8")
12    w.Write([]byte(html))
13 }
14
15 func main() {
16     http.HandleFunc("/", HomeRouterHandler)
17     http.HandleFunc("/another_page", AnotherRouterHandler)
18
19     err := http.ListenAndServe(":3000", nil)
20     if err != nil {
21         log.Fatal("ListenAndServe: ", err)
22     }
23 }
```

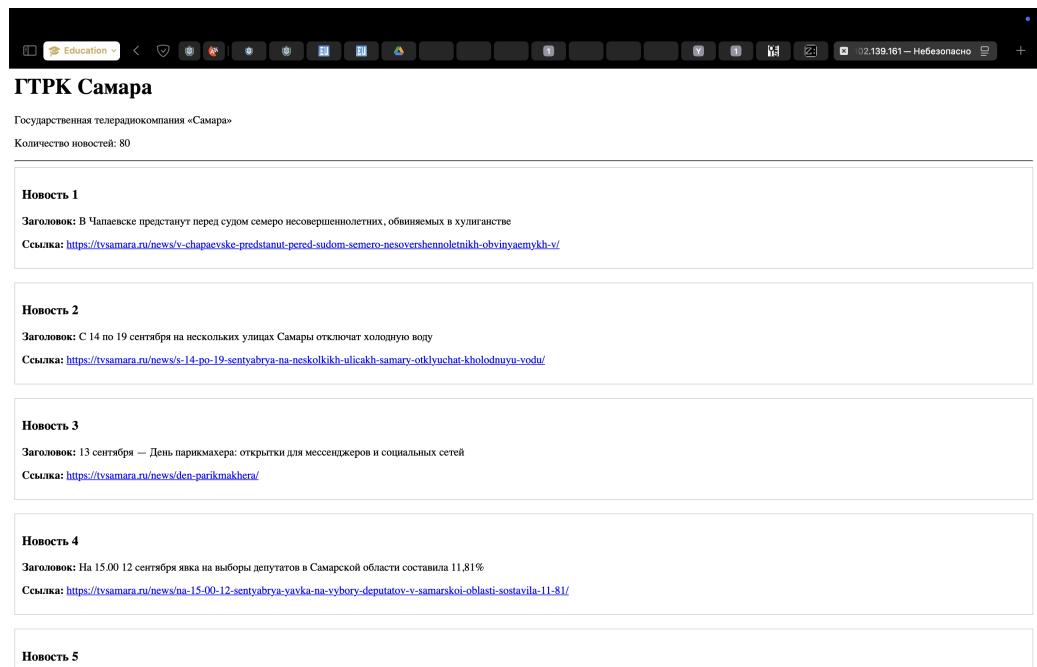


Рис. 1 — Результат работы

3 Вывод

Я научился реализовывать сервер на GO