



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 № 1**  
**по курсу «Компьютерные сети»**  
**«web-сервера»**

Студент группы ИУ9-32Б Волохов А. В.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Задание

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

- Реализовать web-сервер и запустить на заданном порте.
- Изучить принимаемые web-сервером параметры, реализовать передачу данных методом POST. Вывести значения введенные в форму
- Реализовать вывод форматированного гипертекста с контекстным меню в виде гиперссылок, при клике на гиперссылку должна выполняться подмена контента;

Далее рассматривается задача разработки приложения на языке GO реализующего синтаксический разбор XML файла формата RSS. Необходимо Реализовать получение данных из различных RSS- каналов по вариантам. Сравнить результаты разбора и сделать выводы. В итоге необходимо разработать web-сервер, который выполняет соединение с удаленным (удаленными) серверами RSS-новостей и возвращает результаты обработки данных в структурированном виде (страница гипертекста) web-клиенту, в нашем случае в браузер по вариантам.

## 2 Результаты

Исходный код программы представлен в листингах 1– 2– 3.

## Листинг 1 — o.1.go

```
1 package main
2
3 import (
4     "fmt"
5     "html/template"
6     "io/ioutil"
7     "net/http"
8     "sync"
9 )
10
11 var data struct {
12     Name string
13     Age  string
14 }
15
16 var mutex sync.Mutex
17
18 func FormHandler(w http.ResponseWriter, r *http.Request) {
19     if r.Method == http.MethodPost {
20         name := r.FormValue("name")
21         age := r.FormValue("age")
22
23         mutex.Lock()
24         data.Name = name
25         data.Age = age
26         mutex.Unlock()
27
28         http.Redirect(w, r, "/data", http.StatusSeeOther)
29         return
30     }
31
32     w.Header().Set("Content-Type", "text/html")
33     http.ServeFile(w, r, "/home/alex/BMSTU_git/IU9-CN-GO/lab1/0.1/form.
34         html")
35 }
36
37 func DataHandler(w http.ResponseWriter, r *http.Request) {
38     mutex.Lock()
39     defer mutex.Unlock()
```

## Листинг 2 — o.1.go - продолжение

```
1  htmlContent, err := ioutil.ReadFile("/home/alex/BMSTU_git/IU9-CN-GO/
   lab1/0.1/data.html")
2  if err != nil {
3      http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
4      return
5  }
6
7  tmpl, err := template.New("data").Parse(string(htmlContent))
8  if err != nil {
9      http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
10     return
11 }
12
13 w.Header().Set("Content-Type", "text/html")
14 err = tmpl.Execute(w, data)
15 if err != nil {
16     http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
17     return
18 }
19 }
20
21 func main() {
22     http.HandleFunc("/", FormHandler)
23     http.HandleFunc("/data", DataHandler)
24
25     err := http.ListenAndServe(":9000", nil)
26     if err != nil {
27         fmt.Println("Server error:", err)
28     }
29 }
```

## Листинг 3 — form.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Form</title>
6 </head>
7 <body>
8 <h1>Enter the data:</h1>
9 <form action="/" method="post">
10     <label for="name">Name:</label>
11     <input type="text" id="name" name="name"><br><br>
12
13     <label for="age">Age:</label>
14     <input type="text" id="age" name="age"><br><br>
15
16     <input type="submit" value="Send">
17 </form>
18 </body>
19 </html>
```

## Листинг 4 — data.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Form data</title>
6 </head>
7 <body>
8 <h1>Form data:</h1>
9 <p>Name: {{.Name}}</p>
10 <p>Age: {{.Age}}</p>
11 </body>
12 </html>
```

## Листинг 5 — o.2.go

```
1 package main
2
3 import (
4     "fmt"
5     "github.com/mmcdoyle/gofeed"
6     "log"
7 )
8
9 func main() {
10     fp := gofeed.NewParser()
11
12     rssUrls := []string{
13         "https://vesti-k.ru/rss/",
14     }
15
16     for _, url := range rssUrls {
17         feed, err := fp.ParseURL(url)
18         if err != nil {
19             log.Fatalf("Error parsing RSS from URL %s: %v\n", url, err)
20         }
21
22         fmt.Printf("Title : %s\n", feed.Title)
23         fmt.Printf("Description : %s\n", feed.Description)
24         fmt.Printf("Number of Items : %d\n", len(feed.Items))
25
26         for i, item := range feed.Items {
27             fmt.Println()
28             fmt.Printf("Item Number : %d\n", i)
29             fmt.Printf("Title : %s\n", item.Title)
30             fmt.Printf("Link : %s\n", item.Link)
31             fmt.Printf("Description : %s\n", item.Description)
32             fmt.Printf("Published Date : %s\n", item.Published)
33         }
34     }
35 }
```

## Листинг 6 — o.3.go

```
1 package main
2
3 import (
4     "fmt"
5     "github.com/mmcdoyle/gofeed"
6     "html/template"
7     "net/http"
8     "sync"
9 )
10
11 var data struct {
12     RssUrl string
13 }
14
15 var mutex sync.Mutex
16
17 func FormHandler(w http.ResponseWriter, r *http.Request) {
18     if r.Method == http.MethodPost {
19         rssUrl := r.FormValue("rssUrl")
20
21         mutex.Lock()
22         data.RssUrl = rssUrl
23         mutex.Unlock()
24
25         http.Redirect(w, r, "/news", http.StatusSeeOther)
26         return
27     }
28
29     w.Header().Set("Content-Type", "text/html")
30     http.ServeFile(w, r, "/home/alex/BMSTU_git/IU9-CN-GO/lab1/0.3/form.
31         html")
32 }
33
34 func NewsHandler(w http.ResponseWriter, r *http.Request) {
35     mutex.Lock()
36     rssUrl := data.RssUrl
37     mutex.Unlock()
38
39     fp := gofeed.NewParser()
40     feed, err := fp.ParseURL(rssUrl)
41     if err != nil {
42         http.Error(w, "Error parsing RSS feed", http.
43             StatusInternalServerError)
44         return
45     }
46 }
```

## Листинг 7 — o.3.go - продолжение

```
1
2  tmpl, err := template.ParseFiles("/home/alex/BMSTU_git/IU9-CN-GO/lab1
3    /0.3/news.html")
4  if err != nil {
5      http.Error(w, "Error loading template", http.
6        StatusInternalServerError)
7      return
8  }
9  w.Header().Set("Content-Type", "text/html")
10 err = tmpl.Execute(w, feed)
11 if err != nil {
12     http.Error(w, "Error executing template", http.
13       StatusInternalServerError)
14     return
15 }
16 }
17
18 func main() {
19     http.HandleFunc("/", FormHandler)
20     http.HandleFunc("/news", NewsHandler)
21
22     err := http.ListenAndServe(":9000", nil)
23     if err != nil {
24         fmt.Println("Server error:", err)
25     }
26 }
```

## Листинг 8 — form.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Entering an RSS Feed</title>
6 </head>
7 <body>
8 <h1>Enter RSS Feed URL</h1>
9 <form action="/parse" method="post">
10     <label for="rssUrl">RSS feed URL:</label>
11     <input type="text" id="rssUrl" name="rssUrl" required>
12     <input type="submit" value="Send">
13 </form>
14 </body>
15 </html>
```

## Листинг 9 — news.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Links to news</title>
6 </head>
7 <body>
8 <h1>Links to news:</h1>
9 <ul>
10   {{range .Items}}
11     <li><a href="{{.Link}}">{{.Title}}</a></li>
12   {{end}}
13 </ul>
14 </body>
15 </html>
```

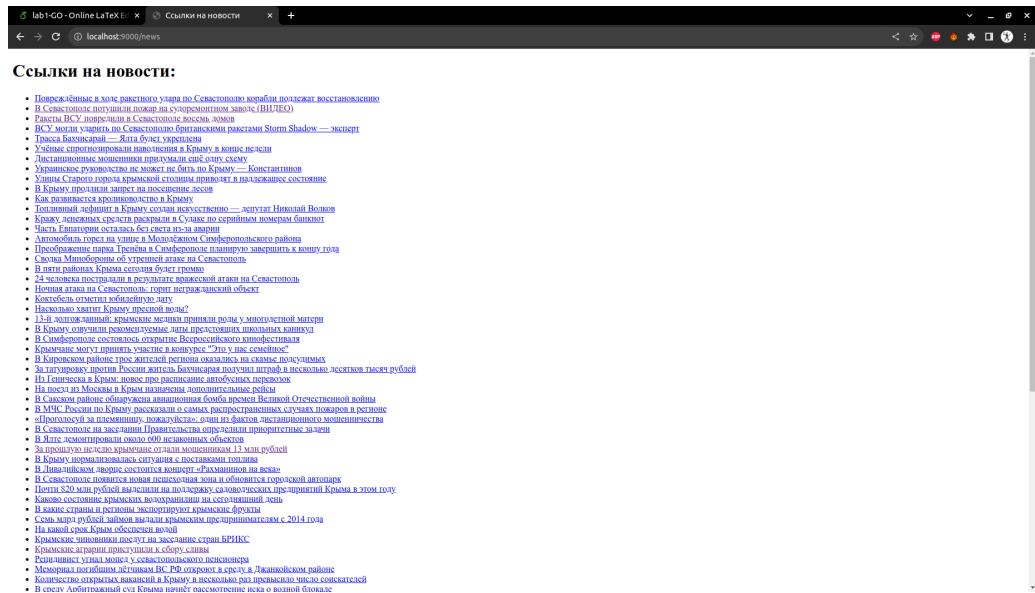


Рис. 1 — Результат