



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 № 9**  
**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 //
12 var data struct {
13     Name string
14     Age  string
15 }
16
17 //
18
19 var mutex sync.Mutex
20
21 func FormHandler(w http.ResponseWriter, r *http.Request) {
22     if r.Method == http.MethodPost {
23         //
24         name := r.FormValue("name")
25         age := r.FormValue("age")
26         //
27
28         mutex.Lock()
29         data.Name = name
30         data.Age = age
31         mutex.Unlock()
32         //
33         http.Redirect(w, r, "/data", http.StatusSeeOther)
34         return
35     }
36     //
37     w.Header().Set("Content-Type", "text/html")
38     http.ServeFile(w, r, "/home/alex/BMSTU_git/IU9-CN-GO/lab1/0.1/form.html")
39 }
40
41 func DataHandler(w http.ResponseWriter, r *http.Request) {
42     //
43
44     mutex.Lock()
45     defer mutex.Unlock()
```

## Листинг 2 — o.1.go - продолжение

```
1 //                                "data.html"
2  htmlContent, err := ioutil.ReadFile("/home/alex/BMSTU_git/IU9-CN-GO/
   lab1/0.1/data.html")
3  if err != nil {
4      http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
5      return
6  }
7
8  //                                "data.html"
9  tmpl, err := template.New("data").Parse(string(htmlContent))
10 if err != nil {
11     http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
12     return
13 }
14
15 //
16 w.Header().Set("Content-Type", "text/html")
17 err = tmpl.Execute(w, data)
18 if err != nil {
19     http.Error(w, "Internal Server Error", http.
        StatusInternalServerError)
20     return
21 }
22 }
23
24 func main() {
25     http.HandleFunc("/", FormHandler)
26     http.HandleFunc("/data", DataHandler)
27
28     err := http.ListenAndServe(":9000", nil)
29     if err != nil {
30         fmt.Println("                :", err)
31     }
32 }
```

### Листинг 3 — form.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>           </title>
6 </head>
7 <body>
8 <h1>               :</h1>
9 <form action="/" method="post">
10  <label for="name">      :</label>
11  <input type="text" id="name" name="name"><br><br>
12
13  <label for="age">       :</label>
14  <input type="text" id="age" name="age"><br><br>
15
16  <input type="submit" value="           ">
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>           </title>
6 </head>
7 <body>
8 <h1>               :</h1>
9 <p>      : {{.Name}}</p>
10 <p>      : {{.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     //                                RSS
11     fp := gofeed.NewParser()
12
13     //                                URL-                                RSS-                                ,
14     rssUrls := []string{
15         "https://vesti-k.ru/rss/",
16         //                                URL-                                RSS-                                ,
17     }
18
19     for _, url := range rssUrls {
20         //                                RSS-
21         feed, err := fp.ParseURL(url)
22         if err != nil {
23             log.Fatalf("                                RSS                                URL %s: %v\n", url, err)
24         }
25
26         //
27         fmt.Printf("Title : %s\n", feed.Title)
28         fmt.Printf("Description : %s\n", feed.Description)
29         fmt.Printf("Number of Items : %d\n", len(feed.Items))
30
31         //
32         for i, item := range feed.Items {
33             fmt.Println()
34             fmt.Printf("Item Number : %d\n", i)
35             fmt.Printf("Title : %s\n", item.Title)
36             fmt.Printf("Link : %s\n", item.Link)
37             fmt.Printf("Description : %s\n", item.Description)
38             fmt.Printf("Published Date : %s\n", item.Published)
39         }
40     }
41 }

```

## 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 //
12 var data struct {
13     RssUrl string
14 }
15
16 //
17 var mutex sync.Mutex
18
19 func FormHandler(w http.ResponseWriter, r *http.Request) {
20     if r.Method == http.MethodPost {
21         // URL RSS-
22         rssUrl := r.FormValue("rssUrl")
23
24         // URL
25
26         mutex.Lock()
27         data.RssUrl = rssUrl
28         mutex.Unlock()
29
30         // /news
31         http.Redirect(w, r, "/news", http.StatusSeeOther)
32         return
33     }
34     // URL RSS-
35
36     w.Header().Set("Content-Type", "text/html")
37     http.ServeFile(w, r, "/home/alex/BMSTU_git/IU9-CN-GO/lab1/0.3/form.html")
38 }
39
40 func NewsHandler(w http.ResponseWriter, r *http.Request) {
41     //
42
43     mutex.Lock()
44     rssUrl := data.RssUrl
45     mutex.Unlock()
46
47     // RSS-
48     fp := gofeed.NewParser()
49     feed, err := fp.ParseURL(rssUrl)
50     if err != nil {
51         http.Error(w, "RSS-", http.StatusInternalServerError)
52     }
53     return
54 }

```

## Листинг 7 — o.3.go - продолжение

```

1
2  //
3      news.html
4  tmpl, err := template.ParseFiles("/home/alex/BMSTU_git/IU9-CN-GO/lab1
5      /0.3/news.html")
6  if err != nil {
7      http.Error(w, "
8      http.StatusInternalServerError)
9      return
10 }
11 w.Header().Set("Content-Type", "text/html")
12 err = tmpl.Execute(w, feed)
13 if err != nil {
14     http.Error(w, "
15     ", http.StatusInternalServerError)
16     return
17 }
18 }
19 }
20 func main() {
21     http.HandleFunc("/", FormHandler)
22     http.HandleFunc("/news", NewsHandler) //
23     "/news"
24
25     err := http.ListenAndServe(":9000", nil)
26     if err != nil {
27         fmt.Println("
28         : ", err)
29     }
30 }

```

## Листинг 8 — form.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>
6     </title>
7 </head>
8 <body>
9 <h1>
10 :</h1>
11 <form action="/" method="post">
12 <label for="name">
13 :</label>
14 <input type="text" id="name" name="name"><br><br>
15
16 <label for="age">
17 :</label>
18 <input type="text" id="age" name="age"><br><br>
19
20 <input type="submit" value="
21 ">
22 </form>
23 </body>
24 </html>

```



## ЛИСТИНГ 9 — news.html

```

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

```

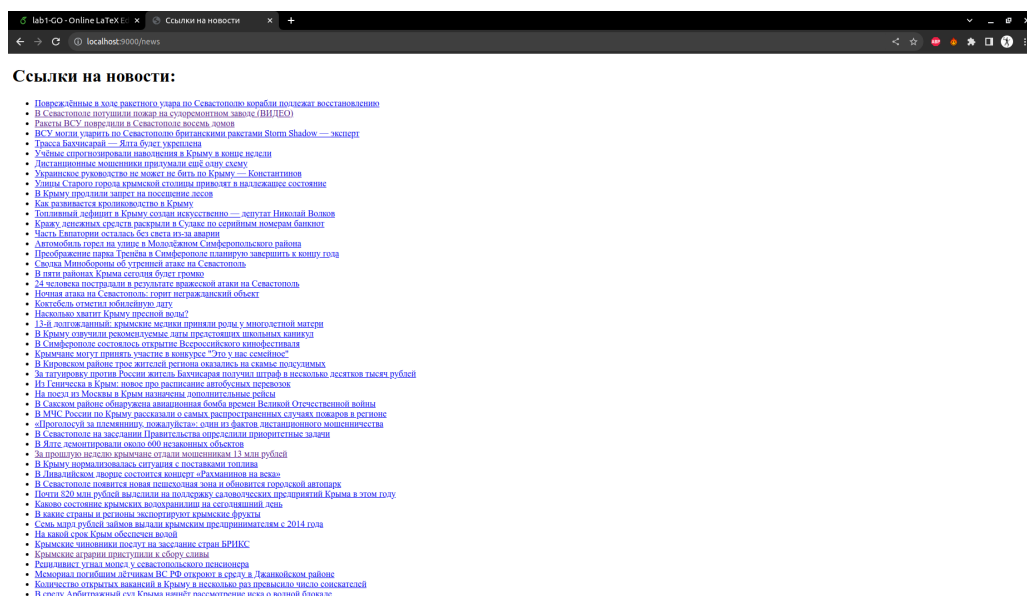


Рис. 1 — Результат