

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 № 0.1 Вариант №28.
Разработка простейшего web-сервера Go

Выполнил
студент группы ИУ9-31Б
Краев Степан

Москва, 2023

1 Условие

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

2 Код решения

Файл server.go

```
package main

import (
    "encoding/xml"
    "fmt"
    "html/template"
    "io/ioutil"
    "log"
    "net/http"
    "path/filepath"
)

type RSS struct {
    Channel Channel `xml:"channel"`
}

type Channel struct {
    Title      string `xml:"title"`
    Description string `xml:"description"`
    Items      []Item `xml:"item"`
}

type Item struct {
    Title      string `xml:"title"`
    Link       string `xml:"link"`
    Description string `xml:"description"`
    Category   string `xml:"category"`
    PubDate    string `xml:"pubDate"`
    GuID       string `xml:"guid"`
}

func getRSSData() (RSS, error) {
    resp, err := http.Get("http://static.feed.rbc.ru/rbc/logical/footer/news.rbc.ru/rss")
    if err != nil {
        return RSS{}, err
    }
    defer resp.Body.Close()
```

```

    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        return RSS{}, err
    }

    var rss RSS
    err = xml.Unmarshal(body, &rss)
    if err != nil {
        return RSS{}, err
    }

    return rss, nil
}

func HomeRouterHandler(rw http.ResponseWriter, r *http.Request) {
    //указываем путь к нужному файлу
    path := filepath.Join("public", "html", "home.html")
    //создаем html-шаблон
    tpl, err := template.ParseFiles(path)
    if err != nil {
        http.Error(rw, err.Error(), 400)
        return
    }
    //выводим шаблон клиенту в браузер
    err = tpl.Execute(rw, nil)
    if err != nil {
        http.Error(rw, err.Error(), 400)
        return
    }
}

func wayout(w http.ResponseWriter, r *http.Request) {
    http.Redirect(w, r, "https://iu9.yss.su/main/lab_id132", http.StatusSeeOther)
}

func about(rw http.ResponseWriter, r *http.Request) {
    path := filepath.Join("public", "html", "empty.html")
    //создаем html-шаблон

```

```

    tmpl, err := template.ParseFiles(path)
    if err != nil {
        http.Error(rw, err.Error(), 400)
        return
    }
    //выводим шаблон клиенту в браузер
    err = tmpl.Execute(rw, nil)
    if err != nil {
        http.Error(rw, err.Error(), 400)
        return
    }
}

func RssRouterHandler(w http.ResponseWriter, r *http.Request) {
    rss, err := getRSSData()
    if err != nil {
        http.Error(w, "Error fetching RSS data", http.StatusInternalServerError)
        return
    }

    fmt.Fprintf(w, "<h1>%s</h1>\n", rss.Channel.Title)
    fmt.Fprintf(w, "<p>%s</p>\n", rss.Channel.Description)

    for _, item := range rss.Channel.Items {
        fmt.Fprintf(w, "<h2><a href='%s'>%s</a></h2>\n", item.Link, item.Title)
        fmt.Fprintf(w, "<p>%s</p>\n", item.Description)
        fmt.Fprintf(w, "<p>Category: %s</p>\n", item.Category)
        fmt.Fprintf(w, "<p>Published Date: %s</p>\n", item.PubDate)
        fmt.Fprintf(w, "<p>GuID: %s</p>\n", item.GuID)
    }
}

func main() {
    http.HandleFunc("/", HomeRouterHandler) // установим роутер
    http.HandleFunc("/way", wayout)
    http.HandleFunc("/description", about)
    http.HandleFunc("/aboutus", RssRouterHandler)
    err := http.ListenAndServe(":9000", nil) // задаем слушать порт
    if err != nil {
        log.Fatal("ListenAndServe: ", err)
    }
}

```

```
}  
}
```

3 Пример работы программы

Приветствую тебя на стартовой странице этого сайта!

- [thewayout](#)
- [opisaniye](#)
- [about rss](#)

Рис. 1: Вывод программы