

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №8
«Разработка универсального web-сервера»
по курсу: «Компьютерные сети»

Выполнил:
Студент группы ИУ9-32Б
Гнатенко Т. А.

Проверил:
Посевин Д. П.

Москва, 2022

Цели

Рассматривается задача разработки универсального web-сервера, который способен обрабатывать результаты выполнения программ на различных языках программирования или результаты работы интерпретаторов по вариантам и передавать их клиенту, а также изображения в формате jpg, png,gif; текстовые файлы, html-документы. Загрузка файлов на веб-сервер должна выполняться с помощью ftp сервера или ssh сервера по вариантам. Должны быть реализованы методы GET и POST. Форматы передачи параметров методом GET: `http://host/some/path/app.exe?a=1&b=2&c=3` или `http://host/some/path/app/1/2/3`.

Задачи

Язык: Java Сервер: FTP

Решение

Исходный код

`server.go`

```
package main
```

```
import (  
    "bytes"  
    "flag"  
    "net/http"  
    "os/exec"  
    "strconv"  
    "strings"  
  
    log "github.com/mgutz/logxi/v1"  
  
    filedriver "github.com/goftp/file-driver"  
    "github.com/goftp/server"  
)
```

```

func runFTPServer() {
    var (
        root = flag.String("root", "static", "Root directory
↳ to serve")
        user = flag.String("user", "user", "Username for
↳ login")
        pass = flag.String("pass", "123456", "Password for
↳ login")
        port = flag.Int("port", 2121, "Port")
        host = flag.String("host", "localhost", "Host")
    )
    flag.Parse()
    if *root == "" {
        log.Error("Please set a root to serve with -root")
    }

    factory := &filedriver.FileDriverFactory{
        RootPath: *root,
        Perm:      server.NewSimplePerm("user", "group"),
    }

    opts := &server.ServerOpts{
        Factory:  factory,
        Port:     *port,
        Hostname: *host,
        Auth:     &server.SimpleAuth{Name: *user, Password:
↳ *pass},
    }

    log.Info("Starting ftp server on", "host: ",
↳ opts.Hostname, "port: ", opts.Port)
    log.Info("Connection with: ", "name: ", *user,
↳ "password: ", *pass)
    server := server.NewServer(opts)
    err := server.ListenAndServe()
    if err != nil {
        log.Error(err.Error())
    }
}

```

```

func goJava(nameFile string, params []string) string {

    var bufIn, bufOut bytes.Buffer
    cmd := exec.Command("javac", nameFile+".java")
    cmd.Dir = "static/Java/"
    if err := cmd.Run(); err != nil {
        log.Error(err.Error())
    }

    bufIn.WriteString(strings.Join(params, "\n"))

    cmd = exec.Command("java", nameFile)
    cmd.Dir = "static/Java/"
    cmd.Stdin = &bufIn
    cmd.Stdout = &bufOut

    if err := cmd.Run(); err != nil {
        log.Error(err.Error())
    }

    return bufOut.String()
}

func handleJava(w http.ResponseWriter, r *http.Request) {
    var params []string
    i := 1

    for par := r.FormValue("p" + strconv.Itoa(i)); par !=
        ↪ ""; par = r.FormValue("p" + strconv.Itoa(i)) {
        params = append(params, par)
        i++
    }

    file := r.RequestURI
    i = strings.LastIndex(file, ".java")
    file = file[6:i]

    w.Write([]byte(goJava(file, params)))
}

```

```

}

func main() {
    go runFTPServer()

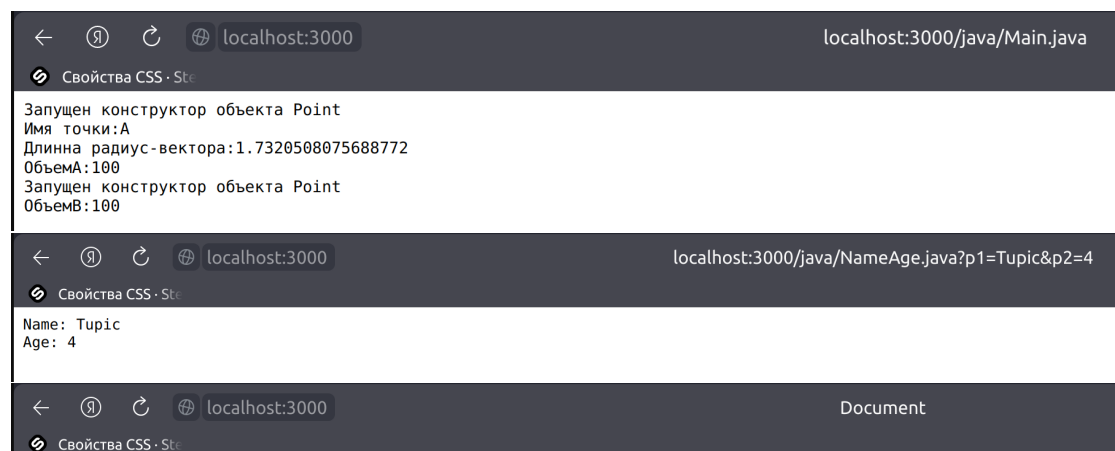
    fs := http.FileServer(http.Dir("./static"))
    http.Handle("/", fs)
    http.HandleFunc("/java/", handleJava)

    log.Info("Listening on :3000")

    err := http.ListenAndServe(":3000", nil)
    if err != nil {
        log.Error(err.Error())
    }
}
}

```

Вывод



Тупик!