

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 №7
«Разработка smtp клиента и приложения почтовой рассылки»
по курсу: «Компьютерные сети»

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

Проверил:
Посевин Д. П.

Москва, 2022

Цели

Рассматривается задача разработки smtp-клиента на языке Golang. Используя пакеты <https://pkg.go.dev/net/smtp>, <https://pkg.go.dev/crypto/tls>, а также в зависимости от реализации возможно понадобится <https://pkg.go.dev/strings>. Необходимо реализовать задачи приведенные ниже.

Задачи

Задача 1: SMTP-клиент на Golang. Необходимо реализовать программу отправки проверочного SMTP сообщения, которое необходимо производить на ящик danila@posevin.com со своего ящика, используемый для переписки с преподавателем. Время и дата получения письма является временем и датой сдачи задания. При этом работоспособность приложения необходимо продемонстрировать очно на следующей лабораторной работе. В этом приложении должны быть реализованы следующие функции: 1. ввод значения поля To из командной строки; 2. ввод значения поля Subject из командной строки; 3. ввод сообщения в поле Message body из командной строки. 4. имя пользователя, которому отправляется сообщение. При отправке проверочного сообщения необходимо в теме сообщения обязательно указать фамилию, имя и группу студента выполнившего задание.

Задача 2: Доработка SMTP-клиента.

1. В базе данных MySQL создать таблицу логирования кодов ответа SMTP сервера (см. <https://hoster.ru/articles/oshibki-smtp-servera-i-sposoby-ihresheniya>) после отправки сообщения приложением написанным в Задаче 1. Поля таблицы логирования следующие — адрес получателя, тема сообщения, текст сообщения, имя получателя письма, код ответа SMTP сервера, расшифровка ответа SMTP сервера.
2. Доработать приложение из Задачи 1 так, чтобы текст письма был оформлен в HTML формате, при этом как минимум приветствие должно быть выделено жирным шрифтом, текст письма курсивом и фон письма отличаться от белого, рекомендуется прочитать статьи приведенные ниже о верстке электронных писем для рассылок.
3. Доработать Задачу 1 так, чтобы была возможность прикрепить один или более вложенных файлов к письму
4. Доработать Задачу 1 следующим образом: если при отправке письма на адрес электронной почты произошла ошибка 550 (см. <https://hoster.ru/articles/oshibki-smtp->

servera-i-sposoby-ih-resheniya), то предлагать пользователю отправить письмо на другой адрес электронной почты. 5. Работоспособность приложения необходимо продемонстрировать очно на следующей лабораторной работе. Параметры доступа к базе данных: □ adminer: <http://students.yss.su/adminer/> □ host: students.yss.su □ db: iu9networkslabs □ login: iu9networkslabs □ passwd: Je2dTYr6

Решение

Исходный код

client.go

```
package main

import (
    "bufio"
    "bytes"
    "database/sql"
    "encoding/base64"
    "fmt"
    "io/ioutil"
    "net/smtp"
    "os"
    "strings"
    "text/template"

    _ "github.com/go-sql-driver/mysql"
    log "github.com/mgutz/logxi/v1"
)

type lett struct {
    Name      string
    Message   string
}

type Mail struct {
    Sender    string
    To        []string
}
```

```

    Subject string
    Body    string
}

func BuildMail(mail Mail) []byte {

    var buf bytes.Buffer

    buf.WriteString(fmt.Sprintf("From: %s\r\n",
↪ mail.Sender))
    buf.WriteString(fmt.Sprintf("To: %s\r\n",
↪ strings.Join(mail.To, ";")))
    buf.WriteString(fmt.Sprintf("Subject: %s\r\n",
↪ mail.Subject))

    fmt.Println("Do you want to attach a file (if so, enter
↪ a name, else enter 'no')?")
    files, err := ioutil.ReadDir("./")
    if err != nil {
        log.Error(err.Error())
    }

    for _, f := range files {
        fmt.Println(f.Name())
    }

    var NamesFile []string
    var fileName string
    fmt.Scanln(&fileName)
    for fileName != "no" {
        NamesFile = append(NamesFile, fileName)
        fmt.Println("More?")
        fmt.Scanln(&fileName)
    }

    boundary := "my-boundary-779"
    buf.WriteString("MIME-Version: 1.0\r\n")
    buf.WriteString(fmt.Sprintf("Content-Type:
↪ multipart/mixed; boundary=%s\n",

```

```

        boundary))

    buf.WriteString(fmt.Sprintf("\r\n--%s\r\n", boundary))
    buf.WriteString("Content-Type: text/html;
↪ charset=\"utf-8\"\r\n")
    buf.WriteString(fmt.Sprintf("\r\n%s\r\n", mail.Body))
    if len(NamesFile) != 0 {
        for _, fileName := range NamesFile {

            data, err := ioutil.ReadFile(fileName)

            if err != nil {
                log.Error(err.Error())
            } else {
                buf.WriteString(fmt.Sprintf("\r\n--%s\r\n",
↪ boundary))
                buf.WriteString("Content-Type: text/plain;
↪ charset=\"utf-8\"\r\n")
                buf.WriteString("Content-Transfer-Encoding:
↪ base64\r\n")
                buf.WriteString("Content-Disposition:
↪ attachment; filename=" + fileName + "\r\n")
                buf.WriteString("Content-ID: <" + fileName +
↪ ">\r\n\r\n")

                b := make([]byte,
↪ base64.StdEncoding.EncodedLen(len(data)))
                base64.StdEncoding.Encode(b, data)
                buf.Write(b)
                log.Info("File is attached success", "file:
↪ ", fileName)
            }
        }

        buf.WriteString(fmt.Sprintf("\r\n--%s", boundary))
        buf.WriteString("--")
    }

    return buf.Bytes()
}

```

```

func main() {
    auth := smtp.PlainAuth("", "t.gnatenko.2003@mail.ru",
        ↪ "df0sc9qYq04kWfg7wkyD",
            "smtp.mail.ru")
    var to string
    fmt.Println("Message To:")
    fmt.Scanln(&to)
    fmt.Println("Message Subject:")
    in := bufio.NewReader(os.Stdin)
    subj, _ := in.ReadString('\n')
    fmt.Println("Message Body:")
    message, _ := in.ReadString('\n')
    fmt.Println("Print Name:")
    name, _ := in.ReadString('\n')
    // to := "tanya-g99@ya.ru" // "danila@posevin.com"
    // message := ""
    // name := "Tupic"
    // subj := ""

    // c, err := smtp.Dial("smtp.mail.ru:25")
    // if err != nil {
    //     log.Error(err.Error())
    // }
    // defer c.Close()
    // err = c.Verify(to)
    // if err != nil {
    //     log.Error(err.Error())
    // }
    body := template.Must(template.New("data").Parse(`
<table bgcolor="#F0FFF0" border="0" cellpadding="0"
    ↪ cellspacing="0"
style="margin:0; padding: 0 10px">
<tr>
<td>
<center style="max-width: 600px; width: 100%;">
<p><b>Здравствуйте, {{.Name}}!</b></p>
<p><i>{{.Message}}</i></p>
</center>
</td>

```

```

</tr>
</table>`))
    buf := new(bytes.Buffer)
    body.Execute(buf, lett{
        Name:    name,
        Message: message,
    })
    request := Mail{
        Sender: "t.gnatenko.2003@mail.ru",
        To:      []string{to},
        Subject: subj,
        Body:     buf.String(),
    }
    msg := BuildMail(request)
    err := smtp.SendMail("smtp.mail.ru:25", auth,
        ↪ "t.gnatenko.2003@mail.ru", []string{to}, msg)
    code := ""
    response := ""
    if err != nil {
        log.Error(err.Error())
        code = (err.Error()[:3])
        response = err.Error()[3 : len(err.Error())-1]

    } else {
        log.Info("sending a mail success!")
    }
    db, err := sql.Open("mysql",
    ↪ "iu9networkslabs:Je2dTYr6@tcp(students.yss.su)/iu9networkslabs")

    if err != nil {
        log.Error(err.Error())
        return
    }
    defer db.Close()
    _, err = db.Exec("insert into `lab7` (`Адрес электронной
    ↪ почты получателя`, `Тема сообщения`, `Текст сообщения`,
    ↪ `Имя получателя письма`, `код ответа SMTP сервера`,
    ↪ `расшифровка ответа SMTP сервера`) values (?, ?, ?, ?,
    ↪ ?, ?);",
        to, subj, message, name, code, response)

```

```
    if err != nil {  
        log.Error(err.Error())  
    } else {  
        log.Info("add")  
    }  
}
```


Вывод

```
> make send
LOGXI=* LOGXI_FORMAT=pretty,happy go run send2/*.go
Message To:
tanya-g99@ya.ru
Message Subject:
Lab7
Message Body:
Tupic
Print Name:
Tanya
Do you want to attach a file (if so, enter a name, else enter 'no')?
Makefile
client
go.mod
go.sum
send2
tupic.txt
tupic.txt
More?
Makefile
More?
no
20:08:46.085209 INF ~ File is attached success
      file: : tupic.txt
20:08:46.085499 INF ~ File is attached success
      file: : Makefile
20:08:46.748582 INF ~ sending a mail success!
20:08:46.871051 INF ~ add
```

Lab7



t.gnatenko.2003@mail.ru t.gnatenko.2003@mail.ru Сегодня в 20:15
Я >



tupic.txt

Здравствуйте, Таня!

Tupic