



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 No 6.2**

**по курсу «Компьютерные сети»**

**«Разработка SMTP-клиента и приложения почтовой рассылки»**

Студент группы ИУ9-31Б Пишикина М. В.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Задание

Целью данной работы является разработка smtp-клиента на языке Golang.

## 2 Код решения

Файл main1.go

```
package main

import (
    "crypto/tls"
    "fmt"
    "net/smtp"
    "os"
)

func main() {
    var toEmail, subject, messageBody string

    fmt.Print("Введите адрес получателя: ")
    fmt.Scanln(&toEmail)

    fmt.Print("Введите тему письма: ")
    fmt.Scanln(&subject)

    fmt.Print("Введите текст сообщения: ")
    fmt.Scanln(&messageBody)

    smtpHost := "smtp.mail.ru"
    smtpPort := 465
    username := "iu9@bmstu.posevin.ru"
    password := "zfDYeN4tfgTDtEJY3tF7"
```

```

message := fmt.Sprintf("To: %s\r\nSubject: %s\r\n\r\n%s", toEmail, subject,

tlsConfig := &tls.Config{
    InsecureSkipVerify: true,
    ServerName:         smtpHost,
}

conn, err := tls.Dial("tcp", fmt.Sprintf("%s:%d", smtpHost, smtpPort), tlsConf
if err != nil {
    os.Exit(1)
}

// Авторизация на SMTP сервере
auth := smtp.PlainAuth("", username, password, smtpHost)
client, err := smtp.NewClient(conn, smtpHost)
if err != nil {
    os.Exit(1)
}

if err := client.Auth(auth); err != nil {
    os.Exit(1)
}

// Отправка сообщения
if err := client.Mail(username); err != nil {
    os.Exit(1)
}

if err := client.Rcpt(toEmail); err != nil {
    os.Exit(1)
}

w, err := client.Data()

```

```

if err != nil {
    os.Exit(1)
}
defer w.Close()

_, err = w.Write([]byte(message))
if err != nil {
    os.Exit(1)
}

fmt.Println("Письмо отправлено")
client.Quit()
}

```

Файл main2.go

```

package main

import (
    "crypto/tls"
    "database/sql"
    "fmt"
    "log"
    "math/rand"
    "net/smtp"
    "os"
    "time"

    _ "github.com/go-sql-driver/mysql"
)

```

```

type Person struct {
    name string
    email string
    text string
}

func main() {
    db, err := sql.Open("mysql", "iu9networkslabs:Je2dTYr6@tcp(students.yss.su:3
    if err != nil {
        log.Fatal(err)
    }
    defer db.Close()

    err = db.Ping()
    if err != nil {
        log.Fatal(err)
    }
    rows, err := db.Query("select * from iu9pishikina2")
    if err != nil {
        panic(err)
    }
    defer rows.Close()
    person := []Person{}

    for rows.Next() {
        var p Person
        err := rows.Scan(&p.name, &p.email, &p.text)
        if err != nil {
            fmt.Println(err)
            continue
        }
        person = append(person, p)
    }
}

```

```

for _, p := range person {
    fmt.Println("Письмо для: ", p.email)
    randomDelay := time.Duration(rand.Intn(1000)+1000) * time.Millisecond
    time.Sleep(randomDelay)

    smtpHost := "smtp.mail.ru"
    username := "iu9@bmstu.posevin.ru"
    password := "zfDYeN4tfgTDtEJY3tF7"
    smtpPort := 465
    emailTemplate := `
        <body style="background: #eb4034;">
        <p style="font-style: italic;">Добрый день, <strong>%s</strong>
        <p style="font-style: italic;">%s</p>
        </body>
    `
    message := fmt.Sprintf("To: %s\r\nSubject: %s\r\nContent-Type: text/html\r\n\r\n",
        p.name, fmt.Sprintf(emailTemplate, p.name, p.text))

    tlsConfig := &tls.Config{
        InsecureSkipVerify: true,
        ServerName:         smtpHost,
    }

    conn, err := tls.Dial("tcp", fmt.Sprintf("%s:%d", smtpHost, smtpPort), tlsConfig)
    if err != nil {
        os.Exit(1)
    }

    auth := smtp.PlainAuth("", username, password, smtpHost)
    client, err := smtp.NewClient(conn, smtpHost)
    if err != nil {
        os.Exit(1)
    }
}

```

```

    if err := client.Auth(auth); err != nil {
        os.Exit(1)
    }

    if err := client.Mail(username); err != nil {
        os.Exit(1)
    }

    if err := client.Rcpt(p.email); err != nil {
        os.Exit(1)
    }

    w, err := client.Data()
    if err != nil {
        os.Exit(1)
    }
    defer w.Close()

    _, err = w.Write([]byte(message))
    if err != nil {
        os.Exit(1)
    }
}
}

```


### 3 Пример работы программы (отправка на свою почту)

- `marypishykina@MacBook-Pro-Mary laba 7 % go run client.go`  
Введите адрес получателя: `marypaint4433@mail.ru`  
Введите тему письма: `Totle`  
Введите текст сообщения: `hiiiiii`  
Письмо отправлено

○ Totle



15:10, Ср, 13 декабря 2023 г.

От:  `iu9@bmstu.posevin.ru`

[Подробнее](#)

hiiiiii

- `marypishykina@MacBook-Pro-Mary laba 7 % go run sender.go`  
Письмо для: `my0p4live@gmail.com`  
Письмо для: `marypaint4433@mail.ru`



○ Tisssssh



16:43, Ср, 13 декабря 2023 г.



iu9@bmstu.posevin.ru

16:43

Тоже тестит, но на другую почту)))



Спасибо за информацию

Все понятно

Ничего не пришло



iu9@bmstu.posevin.ru

16:41

Тоже тестит, но на другую почту)))

## 4 Вывод

Научилась разрабатывать одноранговые сетевые службы