Go 实现邮箱功能

首先需要安装一个gomail包

go get -v gopkg.in/gomail.v2

### main函数里实现路由

这里使用了gin框架

func main() {

r := gin.Default()

r.LoadHTMLGlob("./view/\*")

r.GET("/home", func(c \*gin.Context) { //设置路由

c.HTML(http.StatusOK,"home.html",nil)

})

r.POST("/home", func(c \*gin.Context) {

var mallTo []string

mallTo = make([]string,1)

mailAddress := c.PostForm("address")

content := c.PostForm("content")

mallTo[0] = mailAddress

err := SendMail(mallTo,"这是一个邮件",content)

if err != nil {

c.String(http.StatusBadRequest,"sendMailErr:",err)

return

}

db := dao.ConnectToMysql()

defer db.Close()

//向数据库表保存数据

\_,err = db.Exec("insert into email(emailAddress,content) values (?,?)",mailAddress,content)

if err != nil {

fmt.Println("insert into mysql err:",err)

return

}

if err != nil {

c.String(http.StatusBadRequest,"saveToMysqlErr:",err)

return

}

c.HTML(http.StatusOK,"sendSuccess.html",nil)

})

r.Run(":9090")//在9090端口运行

}

### SendMail函数实现邮箱功能

func SendMail(mailTo []string, subject string, body string) error {

// 设置邮箱主体

mailConn := map[string]string{

"user": "your email address", //发送人邮箱（邮箱以自己的为准）

"pass": "your email pass", //发送人邮箱的密码，现在可能会需要邮箱

//开启授权密码后在pass填写授权码<注意！>

"host": "smtp.qq.com", //邮箱服务器（此时用的是qq邮箱）

}

m := gomail.NewMessage(

//发送文本时设置编码，防止乱码。 如果txt文本设置了之后还是乱码，那可以将原txt文本在保存时

//就选择utf-8格式保存

gomail.SetEncoding(gomail.Base64),

)

m.SetHeader("From", m.FormatAddress(mailConn["user"], "otherName")) // 添加别名

m.SetHeader("To", mailTo...) // 发送给用户(可以多个)

m.SetHeader("Subject", subject) // 设置邮件主题

m.SetBody("text/html", body) // 设置邮件正文

//一个文件（加入发送一个 txt 文件）：/tmp/foo.txt，我需要将这个文件以邮件附件的方式进行发送，同时指定附件名为：附件.txt

//同时解决了文件名乱码问题

name := "附件.txt"

m.Attach(contentPath,

gomail.Rename(name), //重命名

gomail.SetHeader(map[string][]string{

"Content-Disposition": []string{

fmt.Sprintf(`attachment; filename="%s"`, mime.QEncoding.Encode("UTF-8", name)),

},

}),

)

/\*

创建SMTP客户端，连接到远程的邮件服务器，需要指定服务器地址、端口号、用户名、密码，如果端口号为465的话，

自动开启SSL，这个时候需要指定TLSConfig

\*/

d := gomail.NewDialer(mailConn["host"], 465, mailConn["user"], mailConn["pass"]) // 设置邮件正文

//d.TLSConfig = &tls.Config{InsecureSkipVerify: true}

err := d.DialAndSend(m)

return err

}

### 连接数据库

const(

//用户名:密码@[连接方式](主机名:端口号)/数据库名

userName = "root" //用户名

password = "password" //密码

address = "127.0.0.1:3306"

database = "scfi"

)

func ConnectToMysql() \*sql.DB {

dsn := userName +":"+ password +"@"+"("+ address +")/"+ database db,err := sql.Open("mysql",dsn)

if err != nil {

fmt.Println("数据库连接失败",err)

return nil

}

return db

}

### 附前端

#### home.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<style>

form{

width: 300px;

height: 350px;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%,-50%);

display:flex;

flex-direction: column;

border:1px solid black ;

}

input[type=email]{

margin: 10%;

width: 60%;

}

textarea{

margin-right: 10%;

margin-left: 10%;

width: 80%;

height: 200px;

resize: none;

}

input[type=submit]{

width: 60px;

margin-right: auto;

margin-left: auto;

margin-top: 20px;

}

</style>

</head>

<body>

<form action="/home" method="post" novalidate autocomplete="off">

<!-- <span>表单</span> -->

<input type="email" name="address" placeholder="请输入邮箱地址" id="address">

<textarea cols="8" rows="16" name="content" placeholder="邮件内容" id="content"></textarea>

<input type="submit" value="提交">

</form>

</body>

</html>

#### sendSuccess.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>success</title>

</head>

<body>

<h1>Send Successfully!</h1>

</body>

</html>