

Bài tập

- Cài đặt các chương trình đã minh họa trong bài giảng của chương

289

CHƯƠNG 7

GIAO TIẾP VỚI EMAIL SERVER

290

Nội dung

- Giới thiệu
- Gửi email
- SMTP
- POP3
- System.Web.Mail
- Giao tiếp lập trình ứng dụng mail

291

Giới thiệu

- Email là phần không thể thiếu trong nhiều lĩnh vực: xã hội, thương mại - kinh tế, quốc phòng,...
- Lập trình để quản lý email là một vấn đề quan trọng và cần thiết
- Trước tiên cần phải hiểu rõ cấu trúc, cách thức gửi của 1 email
- Tiếp cận thông qua lập trình mức socket

292

Gửi email

- Mỗi email đều phải có địa chỉ duy nhất, có dạng: <Username>@<domain name>
- Chú ý: <domain name> phải được nhận diện duy nhất trên hệ thống DNS toàn cầu. <Username> chỉ cần duy nhất trong mail server của người nhận.
- Email không phải được chuyển trực tiếp đến người nhận mà phải đi qua mail server của ISP hoặc của công ty sở hữu domain.

293

Gửi email

- Từ đó mail được chuyển tiếp một lần nữa đến mail server của người nhận sau một khoảng thời gian ngắn
- Gửi mail dùng giao thức SMTP
- Để xác định mail server của người nhận, một mẫu tin MX (Mail Exchange) trong DNS phải có để quản lý

294

SMTP

- SMTP dùng để gửi chứ không nhận mail được
- Mọi mail server phải tuân theo chuẩn SMTP để gửi được đến đích (RFC 821)
- Việc cài đặt một mail server đơn giản để thực hành có thể thông qua hướng dẫn trong bộ cài đặt Windows Server hoặc dùng phần mềm của bên thứ 3 như: AspEmail, Mercury, Pegasus,...

295

Cài đặt SMTP

- SMTP dùng TCP port 25
- Trước khi thực hành nên kiểm tra địa chỉ IP của ISP
- Cách kiểm tra tốt nhất là dùng trình telnet

296

Gửi SMTP mail

- Start → Run: gõ lệnh cmd và click OK
- Gõ lệnh: telnet <server name> 25
- Khi client đã kết nối được với server thì server luôn trả lời với nội dung: 220 <lời chúc mừng> <version-number>
- Khởi động phiên làm việc với các lệnh sau:

297

Gửi SMTP mail

hello

Mail From: <địa chỉ email của người gửi>

Rcpt To: <địa chỉ email của người nhận>

Data

<nội dung của email>

.

Quit

298

Lập trình gửi mail bằng .NET

- Khai báo thư viện:
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.IO;

299

Lập trình gửi mail bằng .NET

- Khai báo đối tượng thuộc lớp TcpClient để quản lý kết nối với server:
TcpClient client;
- Nếu dùng IPEndpoint thì ta có thể truy cập vào máy Server POP bằng địa chỉ IP:
IPEndPoint iep = new
IPEndPoint(IPAddress.Parse(tbserver.Text),
int.Parse(tbport.Text));
TcpClient client = new TcpClient();
client.Connect(iep);

300

Lập trình gửi mail bằng .NET

- Nếu kết nối trực tiếp, cách thức như sau:
client = new TcpClient(tbserver.Text,
int.Parse(tbport.Text));
- Kết nối:
string Data = "Hello";
StreamReader sr = new
StreamReader(client.GetStream());
StreamWriter sw = new
StreamWriter(client.GetStream());
sw.WriteLine(Data);
sw.Flush();

301

Lập trình gửi mail bằng .NET

- Gửi địa chỉ sender cho server:
Data = "MAIL FROM: <" + tbfrom.Text + ">";
sw.WriteLine(Data);
sw.Flush();
- Đọc thông báo gửi về từ server và xử lý nếu cần thiết
- Gửi địa chỉ receiver cho server:
Data = "RCPT TO: <" + tbto.Text + ">";
sw.WriteLine(Data);
sw.Flush();

302

Lập trình gửi mail bằng .NET

- Bắt đầu nội dung thư:
Data = "Data";
sw.WriteLine(Data);
sw.Flush();
- Thiết lập subject:
Data = "SUBJECT:" + tbsub.Text + "\r\n" +
tbnoidung.Text + "\r\n" + "." + "\r\n";
sw.WriteLine(Data);
sw.Flush();

303

Lập trình gửi mail bằng .NET

- Ngắt kết nối:
Data = "QUIT";
sw.WriteLine(Data);
sw.Flush();

304

Lập trình gửi mail bằng .NET

- Đối với SMTP thì chỉ có thể dùng ASCII.GetBytes vì nó là giao thức gửi dựa trên văn bản, dữ liệu nhị phân không được chấp nhận

305

Nhận SMTP mail

- Start → Run: gõ lệnh cmd và click OK
- Gõ lệnh: telnet <server name> 110
- Khi client đã kết nối được với server thì server luôn trả lời với nội dung: 220 <lời chúc mừng> <version-number>
- Khởi động phiên làm việc với các lệnh sau:

306

Nhận SMTP mail

User <địa chỉ email của người nhận>

Pass <mật khẩu của người nhận>

Stat //xem trạng thái của hộp thư

List //liệt kê danh sách thư

Retr <số thứ tự thư muốn nhận>

Dele <số thứ tự thư muốn xóa>

Rset //hủy đánh dấu xóa các thư

Quit

307

Lập trình nhận mail bằng .NET

- Để hoàn thành việc kiểm tra chương trình, ta có thể cài đặt một email reader khác như Microsoft Outlook, cấu hình một vài tài khoản email.
- Thực hiện việc gửi email từ tài khoản email đó
- Kiểm tra kết quả bằng chương trình mới viết

308

POP3

- Post office protocol 3 (POP3) dùng để nhận chứ không gửi email
- Mọi ISP đều có POP3 server và nhiều công ty cung cấp dịch vụ Web hosting cũng hỗ trợ dịch vụ này
- POP3 được định nghĩa trong RFC 1939, hoạt động trên TCP port 110

309

POP3

- POP3 dùng để lưu trữ email thay mặt cho người dùng, sau đó họ có thể lựa chọn để tải về từ server.
- Một số dịch vụ cung cấp không gian đĩa có giới hạn, do đó cần phải có thao tác xóa email
- Một giải pháp thay thế POP3 là Microsoft Exchange

310

Cài đặt POP3

- Giống như SMTP, POP3 cũng là giao thức dựa trên dòng lệnh
- Mỗi dòng lệnh kết thúc bằng ký tự line-feed (enter).
- Khi server hoạt động bình thường, mỗi dòng sẽ bắt đầu với +OK. Nếu lỗi xảy ra, dòng sẽ bắt đầu với -ERR.

311

Cài đặt POP3

- Để truy xuất được mailbox, client phải chứng thực chính nó với username, password
- Client gửi: USER <username><enter>
- Server trả lời: +OK <welcome><enter>
- Client gửi mật khẩu: PASSWORD <password><enter>

312

Cài đặt POP3

- Để xem trạng thái mailbox, client gửi: STAT <enter>
- Server gửi: +OK <số thứ tự> <dung lượng><enter>
- Lấy email chỉ định, client gửi: RETR <số thứ tự>, trong đó <số thứ tự> thuộc danh sách đã liệt kê trong lệnh STAT
- Server gửi: +OK <some message> <enter> <mail body> <enter>.<enter>

313

Cài đặt POP3

- Để xóa email, client gửi: DELE <số thứ tự>
- Server gửi: +OK <some message> <enter>
- Thực sự đến thời điểm này email vẫn chưa xóa mà chỉ ở tình trạng bị đánh dấu xóa
- Để hủy đánh dấu xóa, client gửi: RSET
- Để kết thúc phiên làm việc, đóng kết nối TCP, client dùng lệnh: QUIT <enter>

314

Cài đặt POP3

- Thực hiện khai báo các thư viện:
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.IO;

315

Cài đặt POP3

- Khai báo các đối tượng để kết nối và nhập/xuất dữ liệu:
NetworkStream nStr;
TcpClient tcp;

316

Cài đặt POP3

- Thực hiện kết nối với server:

```
tcp = new TcpClient(txtServer.Text, 110);  
nStr = tcp.GetStream();  
StreamReader sr = new StreamReader(nStr);  
txtNoiDung.Text = sr.ReadLine() + "\r\n";  
MessageBox.Show("Connect succeeded!",  
"Information");
```

317

Cài đặt POP3

- Chứng thực:

```
try {  
    txtNoiDung.Text += SendPop3("USER " + txtUser.Text  
+ "\r\n");  
    txtNoiDung.Text += SendPop3("PASS " + txtPass.Text  
+ "\r\n");  
    MessageBox.Show("Login succeeded!",  
"Information");  
}  
catch (Exception exp) {  
    MessageBox.Show("Error!" + exp.ToString(),  
"Information");  
}
```

318

Cài đặt POP3

- Xem trạng thái mailbox:

```
try
{
    txtNoiDung.Text += SendPop3("STAT " + "\r\n");
    WriteFile(txtNoiDung.Text.ToString());
    MessageBox.Show("Succesed!", "Information");
}
catch (Exception exp)
{
    MessageBox.Show("Error!" + exp.ToString(),
    "Information");
}
```

319

Cài đặt POP3

- Thoát:

```
try
{
    txtNoiDung.Text += SendPop3("QUIT " + "\r\n");
    WriteFile(txtNoiDung.Text.ToString());
    MessageBox.Show("Succesed!", "Information");
}
catch (Exception exp)
{
    MessageBox.Show("Error!" + exp.ToString(),
    "Information");
}
```

320

Cài đặt POP3

- Kết thúc phiên làm việc:

```
try
{
    tcp.Close();
    nStr.Close();
    this.Close();
}
catch (Exception exp)
{
    MessageBox.Show("Error!" + exp.ToString(),
    "Information");
}
```

321

Cài đặt POP3

- Nội dung hàm SendPop3:

```
public string SendPop3(string cmd)
{
    StreamReader sr = new StreamReader(nStr);
    byte[] data = Encoding.ASCII.GetBytes(
    cmd.ToCharArray());
    nStr.Write(data, 0, data.Length);
    return sr.ReadLine() + "\r\n";
}
```

322

Cài đặt POP3

- Nội dung hàm WriteToLogs:

```
public void WriteToLogs(string msg, FileStream
file)
{
    byte[] bData = Encoding.ASCII.GetBytes(
        msg.ToCharArray());
    file.Write(bData, 0, bData.Length);
    file.Flush();
}
```

323

Cài đặt POP3: xóa email

```
TcpClient client = new TcpClient(tbServer.Text, 110);
NetworkStream networkStream = client.GetStream();
StreamReader reader = new
StreamReader(networkStream);
string welcomeMessage = reader.ReadLine();
lstLogs.Items.Add(welcomeMessage);
RunCmd("USER " + tbUser.Text + NEW_LINE,
networkStream);
RunCmd("PASS " + tbPass.Text + NEW_LINE,
networkStream);
RunCmd("DELE 1" + NEW_LINE, networkStream);
RunCmd("QUIT\r\n", networkStream);
```

324

Cài đặt POP3: xóa email

```
private void RunCmd(string cmd, NetworkStream
networkStream)
{
    byte[] bData =
    Encoding.ASCII.GetBytes(cmd.ToCharArray());
    networkStream.Write(bData, 0, bData.Length);
    StreamReader reader = new
    StreamReader(networkStream);
    string retValue = reader.ReadLine();
    WriteToLogs(retValue + NEW_LINE, file);
    lstLogs.Items.Add(cmd.Replace(NEW_LINE, ""));
    lstLogs.Items.Add(retValue);
    lstLogs.Items.Add("");
}
```

325

Bài tập ứng dụng: SPAM filter

- Mục tiêu: chương trình sẽ quét trong mailbox, kiểm email nào chứa văn bản đã chỉ định để thực hiện xóa nó
- Nội dung phần kết nối và chứng thực tương tự như đã viết ở trên. Trình bày ở slide sau là những kỹ thuật mới về cách quét tất cả các mail trong mailbox, tìm kiếm theo nội dung (giả sử “freemoney”)

326

Bài tập ứng dụng: SPAM filter

```
TcpClient clientSocket = new TcpClient( tbServer.Text,
110);
NetworkStream NetStrm = clientSocket.GetStream();
StreamReader RdStrm= new StreamReader(NetStrm);
string Data = sendPOP3("STAT\r\n",NetStrm);
string[] BreakDown = Data.Split(" ".ToCharArray());
int messageCount = Convert.ToInt16(BreakDown[1]);
for (int i=1;i<= messageCount;i++)
{
    StringBuilder message = new StringBuilder("");
    Data = "RETR " + Convert.ToString(i) + "\r\n";
    byte[] szData=
    System.Text.Encoding.ASCII.GetBytes(Data.ToCharArray());
```

327

Bài tập ứng dụng: SPAM filter

```
NetStrm.Write(szData,0,szData.Length);
string szTemp = RdStrm.ReadLine();
while(szTemp!=".") {
    message.Append(szTemp);
    tbStatus.Text += szTemp+"\r\n";
    szTemp = RdStrm.ReadLine();
}
if (message.ToString().IndexOf("free money")>=0) {
    sendPOP3("DELE " + Convert.ToString(i) +
    "\r\n",NetStrm);
}
}
```

328

System.Web.Mail

- System.Web.Mail là công cụ có sẵn trong Windows là được dùng cho việc gửi email.
- Chúng đơn giản hơn SMTP, đặc biệt trong việc đính kèm file và văn bản rich-text
- Ta phải thêm reference đến System.Web.dll trước khi có thể dùng namespace System.Web.Mail

329

System.Web.Mail

```
using System.Web.Mail;
private void btnSend_Click(object sender,
System.EventArgs e)
{
    MailMessage email = new MailMessage();
    email.From = tbFrom.Text;
    email.To = tbTo.Text;
    email.Subject = "email from .NET";
    email.Body = tbMessage.Text;
    SmtpMail.SmtpServer = tbServer.Text;
    SmtpMail.Send(email);
}
```

330

Attachments (Đính kèm)

- Việc đính kèm thêm các đối tượng file khác là điều hiển nhiên phải hỗ trợ trong các email hiện đại
- Để thực hiện việc này chúng ta có thể làm như sau:

331

Attachments (Đính kèm)

```
private void btnSend_Click(object sender,
System.EventArgs e)
{
    MailMessage email = new MailMessage();
    MailAttachment fileAttachment=new
    MailAttachment(tbAttachment.Text);
    email.Priority = MailPriority.High;
    email.BodyFormat = MailFormat.Html;
    email.From = tbFrom.Text;
```

332

Attachments (Đính kèm)

```
email.To = tbTo.Text;  
email.Subject = "email from .NET";  
email.Body = tbMessage.Text;  
email.Attachments.Add(fileAttachment);  
SmtpMail.SmtpServer = tbServer.Text;  
SmtpMail.Send(email);  
}
```

333

Attachments (Đính kèm)

- Đối với các file hình ảnh thì cần thêm một chút xử lý
- Trên web, muốn hiển thị hình ảnh ta dùng: ``, nhưng không thể làm điều đó trên email
- Cách xử lý như sau:
 - Dùng `` trong body email
 - Gọi `attachInlineFile("c:\picture.jpg", "", "picture1")`
 - Gọi hàm Send email

334

Giao tiếp lập trình ứng dụng mail

- Microsoft Outlook có cung cấp 1 interface để các ứng dụng truy xuất email lưu trữ bên trong, được gọi là Mail Application Programming Interface (MAPI), là thư viện kiểu COM (Component Object Model)
- Chúng ta vẫn có thể truy xuất được COM này từ môi trường .NET

335

Giao tiếp lập trình ứng dụng mail

- Để thêm COM trên vào, ta chọn: Project → Add Reference: chọn COM và tìm đến Microsoft Outlook 10.0 Object Library → Select
- Xử lý:
ListViewItem liEmail;
Microsoft.Office.Interop.Outlook.Application App;
Microsoft.Office.Interop.Outlook.MailItem Msg;
Microsoft.Office.Interop.Outlook.NameSpace NS;
Microsoft.Office.Interop.Outlook.MAPIFolder Inbox;
Microsoft.Office.Interop.Outlook.Items Items;
App = new
Microsoft.Office.Interop.Outlook.Application();

336

Giao tiếp lập trình ứng dụng mail

```
NS = App.GetNamespace("mapi");
Inbox = NS.DefaultFolder(
Microsoft.Office.Interop.Outlook.OlDefaultFolders.olFold
erInbox);
Items = Inbox.Items;
for (int i = 1; i < Items.Count; i++)
{
    Msg =
(Microsoft.Office.Interop.Outlook.MailItem)Items[i];
    liEmail = lvOutlook.Items.Add(Msg.SenderName);
    liEmail.SubItems.Add(Msg.Subject);
}
```

337

Network news transfer protocol

- Network news transfer protocol (NNT) chạy trên port 119 và được định nghĩa trong RFC 977
- Giao thức này dùng để quản lý mailing list và hiện nay trở nên lạc hậu
- 2 tác vụ chính:
 - Đọc tin mới
 - Tạo tin mới

338