

## 6.9 应用示例

本节使用 Visual Studio 2010 .NET 开发几个 Web 应用程序实例，这些实例演示如何创建 Web 应用程序、如何在 C# 中使用 ASP.NET 对象等。

### 6.9.1 聊天室

本实例为一个简单的聊天室程序，主页面分为上下两个窗口，上面窗口显示聊天内容，下面窗口显示输入聊天信息的表单，实现基本的网上聊天功能，主要使用的技术为 HTML 框架和 Application 对象。开发聊天室应用程序的步骤如下：

(1) 新建一个名为 chat 的 ASP.NET Web 应用程序：在 VS2010 开发环境中，打开“文件”菜单，选择“新建”|“项目”命令；在“新建项目”对话框中“项目类型”列表框选择“Visual C# 项目”选项，在“模板”列表框中选中“ASP.NET Web 应用程序”模板选项，在“位置”文本框选择本地路径，如图 6.17 所示，然后单击“确定”按钮。

(2) 在资源管理器窗口中，在“chat”项目上点击右键，“添加”新项，添加 Global.asax 文件。如图 6.18 所示。

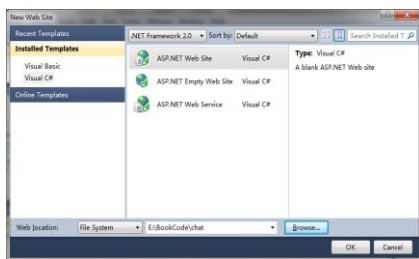


图 6.17 新建 Web 网站



图 6.18 添加 Global.asax 文件

(3) 初始化聊天内容：因为聊天内容要求所有用户都能看到，所以本实例中使用 Application 对象来存储聊天内容。因为读取 Application 中不存在的内容会引起异常，所以要初始化聊天内容。在 Global.asax.cs 代码编辑窗口中，为 Application\_Start() 添加代码如下：

```
protected void Application_Start(Object sender, EventArgs e)
{
    //在应用程序启动时初始化聊天内容
    Application["chatcontent"] = "<h2>欢迎来到聊天室.....</h2>";
}
```

(4) 添加新项 send.aspx：当创建一个新的应用程序时，会默认生成文件 WebForm1.aspx，可以根据需要重新命名。

(5) 设计 send.aspx 的界面：send.aspx 用来让用户输入聊天内容，对应于主页面中下面的窗口。在 send.aspx 的页面开始处输入文本“姓名：”。

从“工具箱”内选择一个 TextBox 控件放在“姓名：”后，并在“属性”窗口中把 TextBox 控件的 ID 属性改为 sender，把 Columns 属性改为 10。

把光标放置在 TextBox 控件 sender 后面，按 Shift+Enter 进行换行。输入文本“内容：”。从“工具箱”内选择一个 TextBox 控件放在“内容：”后，并在“属性”窗口中把 TextBox 控件的 ID 属性改为 content，把 Columns 属性改为 60。

从“工具箱”内选择一个 Button 控件放在 TextBox 控件 content 后，并在“属性”窗口中把 Button 控件的 Text 属性改为“发送”。此时，send.aspx 的界面如图 6.19 所示。

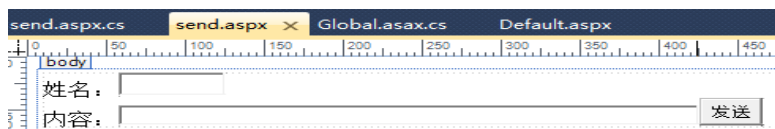


图 6.19 send.aspx 界面

(6) 为 send.aspx 中的按钮添加 Click 事件处理程序：双击 send.aspx 中的按钮，可打开 send.aspx.cs 的代码编辑窗口，光标自动放置在 Button1\_Click()方法内部，此方法会在单击按钮时执行，在此添加代码如下：

```
protected void btnSend_Click(object sender, EventArgs e)
{
    //定义变量,用来存放一条聊天信息,包括说话者、说话内容、说话日期时间
    string message;
    //获取说话者的名字并用蓝色显示
    message="<font color='blue'>" + this.sender.Text + "</font>说:";
    //获取说话内容
    message += this.txtContent.Text;
    //获取说话日期时间并用斜体显示
    message += "(<i>" + DateTime.Now.ToString() + "</i>";
    //在每条信息后面加上换行
    message += "<br>";
    Application.Lock();
    //把新聊天信息附加在原来聊天信息的后面并存入Application
    Application["chatcontent"] = (string)Application["chatcontent"] + message;
    Application.Unlock();
    //清空聊天文本框
    this.txtContent.Text = "";
}
```

(7) 添加 Web 窗体：打开“项目”菜单，选择“添加 Web 窗体”命令，弹出“添加新项”对话框，在“名称”文本框中输入 main.aspx，然后单击“打开”按钮。

(8) 设计 main.aspx 页面：从“工具箱”中拖出一个 Label 控件放在页面上，把 Label 控件的 ID 属性改为 labMessage，把 Text 属性设置为空，然后把 main.aspx 窗口由设计视图切换到 HTML 源视图，并在<head>与</head>之间添加下面代码：

```
<meta http-equiv="refresh" content="4;">
```

这行代码的作用是让页面每 4 秒钟自动刷新一次，以自动更新聊天内容。

(9) 编辑 main.aspx.cs 的代码：打开“视图”菜单，选择“代码”命令，切换到了 main.aspx.cs 的代码编辑窗口。在代码编辑窗口中，为 Page\_Load()添加代码如下：

```
private void Page_Load(object sender, System.EventArgs e)
{
    //把Application中的聊天信息读出来显示在页面中
    this.labMessage.Text = (string)Application["chatcontent"];
}
```

(10) 添加 HTML 页：打开“项目”菜单，选择“添加 HTML 页”命令，弹出“添加新项”对话框，在“名称”文本框中输入 chat.htm，然后单击“打开”按钮。

(11) 编辑 chat.htm：在 chat.htm 的 HTML 源视图窗口中，把源代码修改为：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html><head><title>聊天室欢迎您</title>
<meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
<meta name=ProgId content=VisualStudio.HTML>
<meta name=Originator content="Microsoft Visual Studio .NET 7.1"></head>
<frameset rows="*,80">
    <frame src=main.aspx>
    <frame src=send.aspx noresize>
</frameset>
```

</html>

(12) 把 chat.htm 设为起始页：在“解决方案资源管理器”窗口中，右击文件 chat.htm，在快捷菜单中选择“设为起始页”命令。

(13) 预览应用程序：打开“调试菜单”，选择“启动”命令，也可以直接按 F5 键，就可以看到程序运行的结果，输入自己的网名，就可以聊天了，如图 6.20 所示。

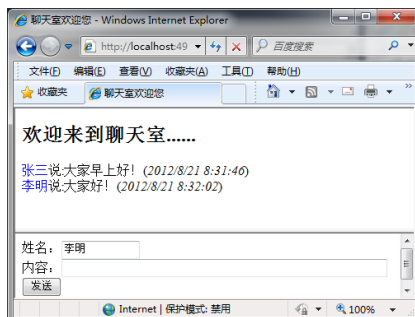


图 6.20 聊天室应用程序运行结果

## 6.9.2 用户权限检查

在 Web 应用程序和网站中，往往存在一些比较特殊的页面，这些页面会自动检查访问者的权限，不同权限的访问者看到的执行结果是不同的。比如，网站的后台管理页面，一般用户是访问不到的，只有登录成功的管理员才能访问，从而执行相应的管理操作。

本例中，用户分为一般用户和管理员两种，主要用 Session 对象来实现。因为本例主要是用来演示如何使用 Session 的，所以页面内容很简单，没有实现具体的管理操作。实例中主要用到三个页面：index.aspx 为起始页，所有用户都可以访问，且看到的内容一样；login.aspx 为登录页面，管理员在本页面进行登录，如果登录成功则自动转到页面 admin.aspx，否则弹出提示信息；admin.aspx 为管理页面，只有管理员才能看到页面内容，一般用户访问时会自动转到登录页面。具体实现过程如下：

(1) 将默认 Default.asp 重命名为 login.aspx，设计页面在表格单元格里输入相应的文本，并从“工具箱”拖出两个 TextBox 控件和一个 Button 控件放入相应单元格中，并设置控件的属性，适当调整表格的大小和列宽。其中两个 TextBox 控件的 ID 属性分别设置为 username 和 userpass，Width 属性都设置为 150px，用于输入密码的 TextBox 控件的 TextMode 属性设置为 Password。Button 控件的 Text 属性设置为“登录”，如图 6.21 所示。

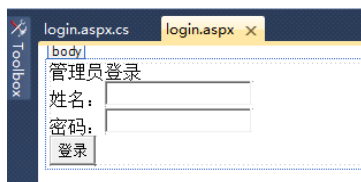


图 6.21 login.aspx 界面设计

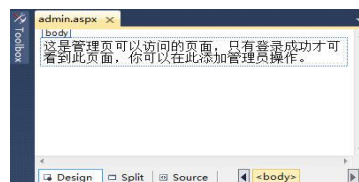


图 6.22 admin.aspx 界面设计

(2) 双击 login.aspx 中的 Button 控件，添加 Click 事件处理程序如下：

```
private void Button1_Click(object sender, System.EventArgs e)
{
    //定义登录失败时弹出信息框的客户端脚本
    string strno="<script>alert('用户名或密码不正确！');</script>";
    //判断是否为管理员，在实际应用中，用户名和密码一般都存在数据库中
    //应先从数据库中读出再进行判断
    if(username.Text=="张三" && userpass.Text=="1234567")
```

```

    { //登录成功，把用户名和密码都存入Session对象中
      Session["username"]=username.Text;
      Session["userpass"]=userpass.Text;
      //把页面转向到admin.aspx
      Response.Redirect("admin.aspx");
    }
    else
    { //登录不成功，并弹出信息
      Page.RegisterClientScriptBlock("loginno",strno);
    }
  }
}

```

(3) 添加 Web 窗体，并命名为 admin.aspx。

(4) 设计 admin.aspx 的界面(如图 6.22 所示)。

(5) 在 admin.aspx 页面空白处双击，打开 admin.aspx.cs 的代码编辑窗口，为 Page\_Load() 添加代码如下：

```

private void Page_Load(object sender, System.EventArgs e)
{ //判断相应的Session信息是否存在
  if(Session["username"]!=null && Session["userpass"]!=null)
  { //相应的Session信息存在，进一步判断是否为管理员
    string username=Session["username"].ToString();
    string userpass=Session["userpass"].ToString();
    if(username!="张三" || userpass!="1234567")
    { //不是管理员，自动转到登录页面并终止执行当前页
      Response.Redirect("login.aspx",true);
    }
  }
  else
  { //相应Session信息不存在，自动转到登录页面并终止执行当前页
    Response.Redirect("login.aspx",true);
  }
}

```

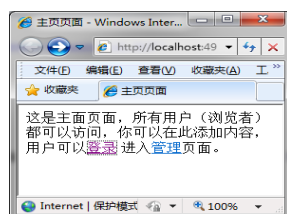


图 6.23 起始页面 index.aspx

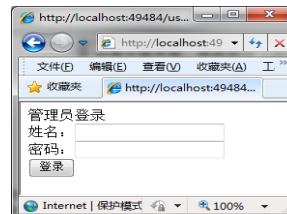
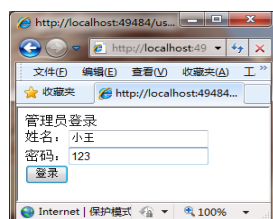


图 6.24 登录页面

(6) 添加 index.aspx 页面，如图 6.23 所示。点击其中的“管理”超链接，虽然链接的目标为 admin.aspx，但打开的却是登录页面 login.aspx，这是因为 admin.aspx 页面程序要检查 Session 变量，若未登录则将页面重定向至 login.aspx。

(7) 在登录页面中，若不输入信息或输入错误的信息，单击“登录”按钮后，会弹出警告信息，如图 6.25 所示，单击警告框的“确定”按钮关闭警告框，将重新显示登录界面。



(a) 输入错误的登录信息



(b) 出错提示

图 6.25 登录失败

(8) 在登录页面中，输入正确的姓名（张三）和密码（1234567），单击“登录”按钮后，浏览器中显示管理页面 `admin.aspx`，如图 6.26 所示。

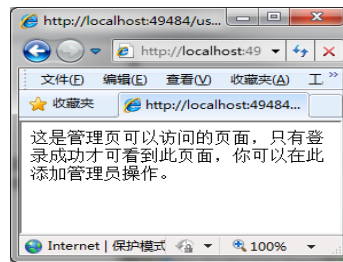


图 6.26 管理页面 `admin.aspx`

本实例实现的功能比较简单，只有一个管理页面，而在实际应用中，可能有多个页面需要管理员权限，在实现时可以采用相同的方法，为每个管理页面添加相同的检查权限代码，当然，也可以用其他的方法实现。