
任务六 实现在线聊天功能



学习目标

通过本任务的学习，使学生了解 Asp.net 内置对象的基本功能，掌握每个对象的常用属性、集合和方法。



知识目标

- 熟悉 ASP.NET 内置对象。
- 掌握 Request 对象的主要功能及基本用法
- 掌握 Response 对象的主要功能及基本用法。
- 掌握 Application 对象的主要功能及基本方法。
- 掌握 Session 对象的主要功能及基本方法。
- 理解 Cookie 的主要功能及基本方法。



技能目标

- 能运用 Resonse 对象和 Request 对象进行页面传值
- 能运用 Request 对象获取客户端浏览器的信息
- 能运用 Application 对象统计在线人数、实现消息发送
- 能运用 Session 对象精确统计在线人数、实现强制登录
- 能运用 Cookie 实现密码记忆功能



项目背景

许多网站都具有在线聊天功能,在 Internet 上的聊天程序一般是以服务器提供连接响应,用户通过客户端程序登录到服务器,就可以与登录同一服务器的其他用户聊天,本任务实现图书馆管理系统的在线聊天功能为已登录的用户提供在线聊天功能,用户登录后可看到用户个人登录信息和当前的在线人数,并能自由发言,查看客户端信息,并且具有密码记忆功能,用户登录一次后,下次登录时输入正确的用户名即可自动获取登录密码进行登录。



任务实施

本模块主要任务是实现图书馆管理系统的在线聊天功能，主要分为用户信息传递、在线留言和统计在线人数三个子任务。

6.1 子任务一 用户信息传递

【任务陈述】

用户登录后，用户信息如用户名和密码等经常需要跨页传递，例如：图书馆管理系统项目的用户登录后，需将用户名、密码等用户信息传递到目标页面。

【知识准备】

6.1.1 五大对象功能概述

Web 应用开发中一个很重要的问题就是 Web 页面之间的信息传递和信息维护，Asp.net 提供了一些内置对象和 Cookie 来管理 Web 页面之间的状态，以实现在页面往返过程中自动保留所有控件的属性值和其他特定值。下表列出了 Asp.net 中的内置对象及其说明。

表 6-1 Asp.net 中的内置对象及其说明

对象名	说明
Response	提供对当前页的输出流的访问。(主要用于生成 HTML 内容并送交浏览器)
Request	提供对当前页请求的访问。(主要用于获取来自客户端的数据，如用户填入表单的数据等)
Application	提供对所有会话的应用程序范围的方法和事件的访问。还提供对可用于存储信息的应用程序范围的缓存的访问。
Session	为当前用户会话提供信息。
Cookies	提供创建和操作各 HTTP Cookies 的类型安全方法。

6.1.2 Response 对象概述

Response 对象用于动态响应客户端请示，控制发送给用户的信息，并动态生成响应，结果返回给客户端浏览器。下面将介绍他的属性和方法。

表 6-2 Response 对象的常用属性和方法

名 称	方法/属性	描 述
BufferOutput	属性	获取或设置一个值，该值指示是否缓冲输出并在处理完整个响应之后发送它
Cookies	属性	获取响应 Cookie 集合
Clear	方法	清除缓冲区流中的所有内容输出
Write	方法	将信息写入 HTTP 响应输出流

WriteFile	方法	将指定的文件直接写入 HTTP 响应输出流
End	方法	将当前所有缓冲的输出发送到客户端，停止该页的执行，并引发 EndRequest 事件
Redirect	方法	将客户端重定向到新的 URL，并指定该新 URL

1. 使用 Write 方法输出信息

Response 对象的 Write 方法用于向客户端输出信息，语法如下：

```
Response.Write(“字符串”|变量);
```

案例：在屏幕上显示“欢迎进入聊天室”。

(1) 新建【WebSite6】网站，添加一个网页，并在网页的 Page_Load 事件中编写如下代码：

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("欢迎来到聊天室");
}
```

(2) 运行结果如图 6-1 所示：



图 6-1 案例运行结果

使用 Write 方法输出的字符串会被浏览器按 HTML 语法进行解释。因此可以使用 Write 方法直接输出 HTML 代码来实现页面内容和格式的定制。

举一反三：分析以下代码的运行结果为什么如图 6-2 所示呢？



图 6-2 案例运行结果

2. BufferOutput 属性以及 Clear 方法

Response 对象的 BufferOutput 属性的默认值为 true, 所以输出到客户端的数据暂时存储在缓冲区内, 等所有的事件程序以及所有的页面对象全部解释完毕后, 才将所有在缓冲区中的数据送到客户端的浏览器, Response 对象的 Clear 方法清除缓冲区流中的所有内容输出, 让我们通过以下两个案例来观察缓冲区如何运作。

案例: 使用缓冲区。

(1) 新建页面, 在网页的 Page_Load 事件中编写如下代码:

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("清除缓冲区之前的数据<br>");
    Response.Clear();
    Response.Write("清除缓冲区之后的数据<br>");
}
```

程序运行后, 首先在页面输出“清除缓冲区之前的数据”这一行, 同时数据已存在缓冲区内, 然后使用 Clear() 方法将缓冲区中的数据清除, 此时, 刚刚送出的数据“清除缓冲区之前的数据”被清除, 最终没有显示“清除缓冲区之前的数据”这一行, 所以程序运行结果如图 6-3 所示。



图 6-3 案例运行结果

案例: 不使用缓冲区。

(1) 新建页面, 在网页的 Page_Load 事件中编写如下代码:

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.BufferOutput = false;
    Response.Write("清除缓冲区之前的数据<br>");
    Response.Clear();
    Response.Write("清除缓冲区之后的数据<br>");
}
```

将 Response 的 BufferOutput 的属性设置为 false, 表示数据没有放在缓冲区内, 而是直接输出, 也没有因为使用 Clear 方法而将缓冲区中的数据清除, 程序运行结果如下:

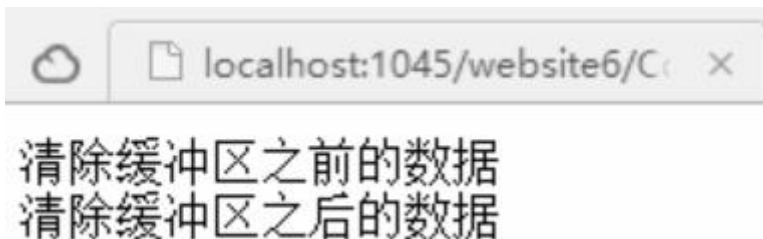


图 6-4 案例运行结果

3. Redirect 方法

Response 对象的 Redirect 方法可以将超链接重新导向其他地址, 使用时只要传入一个 URL 即可。语法如下:

```
Response.Redirect("页面 URL 地址");
Response.Redirect("Page1.htm"); //将网页转移到当前目录的 Page1.htm
Response.Redirect("http://localhost/index.aspx") //将网页转移到"/index.aspx"
```

案例：网页跳转案例。

(1) 新建网页，打开设计视图，输入“请输入 URL”，并添加一【ID】属性为【txtURL】的【TextBox】控件和一个【ID】属性为【BtnSubmit】，【Text】属性为【确定】的【Button】控件。双击【BtnSubmit】控件，并在 BtnSubmit_Click 事件中添加如下代码：

```
protected void BtnSubmit_Click(object sender, EventArgs e)
{
    Response.Redirect(txtURL.Text);
}
```

(2) 程序运行后在文本框中输入 http://www.baidu.com，点击提交后，即可跳转到百度网站。



图 6-6 案例输入演示



图 6-6 案例运行结果

6.1.3 Request 概述

用于获取客户端向服务器端发出的 HTTP 请求中的客户端的信息。Request 对象的属性和方法相当多，下面将介绍它的常用属性和方法。

表 6-3 Request 对象的常用属性和方法

名 称	方法/属性	描 述
Browser	属性	获取有关正在请求的客户端的浏览器功能的信息
Cookies	属性	获取客户端发送的 Cookie 的集合
Form	属性	获取客户端表单元素中所填入的信息
MapPath	方法	为当前请求将请求的 URL 中的虚拟路径映射到服务器上的物理路径
Params	属性	获取 QueryString、Form、ServerVariables 和 Cookies 项的数据
QueryString	属性	获取 HTTP 查询字符串变量集合
SaveAs	方法	将 HTTP 请求保存到磁盘

1. Form 属性的使用方法

服务器获取表单数据的方式取决于客户端表单提交的方式。

(1) 若表单的提交方式为“get”，则表单数据将以字符串形式附加在 URL 之后，在 QueryString 集合中返回服务器。例如：

```
http://localhost/Target.aspx?X1=value1&X2=value2
```

上式中问号“?”之后即为表单中项和数据值：表单项 X1 值为 value1，表单项 X2 值为 value2。

此时，在服务器端要使用 `Request` 对象的 `QueryString` 集合来获取表单数据。例如：

```
Request.QueryString["X1"];           // 获取表单项 XX 的值
Request.QueryString["X2"];           // 获取表单项 YY 的值
```

(2) 若表单的提交方式为“post”，则表单数据将放在浏览器请求的 HTTP 标头中返回服务器，其信息保存在 `Request` 对象的 `Form` 集合中。此时，在服务器端要使用 `Request` 对象的 `Form` 集合来获取表单数据。例如：

```
Request.Form["X1"];                   // 获取表单项 XX 的值
Request.Form["X2"];                   // 获取表单项 YY 的值
```

(3) 无论表单以何种方式提交，都可使用 `Request` 对象的 `Params` 集合来读取表单数据。例如：

```
Request.Params["X1"];                 // 获取表单项 XX 的值
Request.Params["X2"];                 // 获取表单项 YY 的值
```

或者，可以省略 `QueryString`、`Form` 或 `Params`，直接使用形式“`Request[表单项]`”来读取表单数据，例如：

```
Request["X1"];                        // 获取表单项 XX 的值
Request["X2"];                        // 获取表单项 YY 的值
```

案例：设计一个用户登录界面。用户在页面上输入姓名和密码，单击【提交】按钮后利用 `Form` 获取客户端提交的登录信息，并把相关信息显示在页面上。

(1) 在【website6】网站添加一个名为【RequestForm.aspx】的页面，切换到设计视图，输入“姓名：”，并在其后面添加一个【TextBox】控件设置其【ID】属性为【txtName】，换行输入“密码：”，并在其后面添加一个【TextBox】控件设置其【ID】属性为【txtPwd】。换行添加【Button】控件，设置其【Text】属性为【登录】，【ID】属性为【btnSubmit】。

(2) 双击 `Button` 控件，并在添加如下代码：

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    Response.Write("您输入的姓名是：" + Request.Form["txtName"] + "<br>");
    Response.Write("您输入的密码是：" + Request.Form["txtPwd"] + "<br>");
}
```

(3) 运行程序，输入用户名和密码，运行结果如图 6-7 所示。

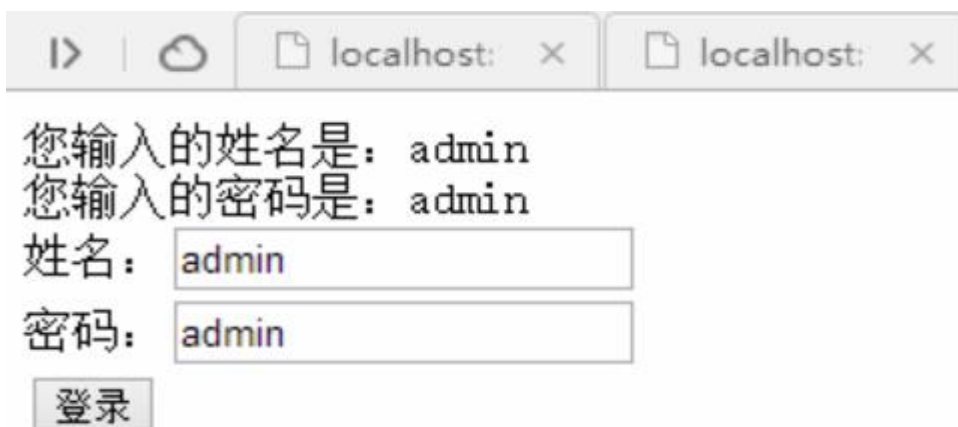


图 6-7 案例运行结果

2. QueryString 属性的使用方法

QueryString 用于检索 url 查询字符串中变量的值。传递变量的名和值由“?”后的内容指定，例如：

网址：http://...../TargetPage.aspx?username="admin"
“?”前面的部分为要访问的网页的 url
“?”后面为相应的 QueryString 变量
username 表示变量的名字，“admin”为其数值，表示向 TargetPage.aspx 页面传递一个名为 username 的变量，值为“admin”，用户可以通过 request.querystring["username"]；获得该数值。

注意：“?”后面可以有多个变量，各个变量之间使用“&”连接。

【实施与测试】

6.1.4 用户信息传递功能实现

(1) 在【Library】网站以【MasterPage.master】为母版添加一个名为【ChatRoom.aspx】的内容页和一个名为【TargetPage.aspx】的网页。

(2) 将 ChatRoom.aspx 切换到设计视图，将相关控件拖入页面的 ContentPlaceHolder1 控件内。

表 6-4 相关控件的选用及设置

功能	控件类型	主要属性设置
面板 1	Panel1	默认属性
姓名	TextBox	Id="txtName" AutoPostBack="True"（放在 Panel1 中）
密码	TextBox	Id="txtPwd" TextMode="Password"（放在 Panel1 中）
记住密码	CheckBox	Id="ckbPwd"（放在 Panel1 中）
登录	Button	Id="btnSubmit" Text="登录"（放在 Panel1 中）
面板 2	Pane2	默认属性
信息显示	Label	默认属性（放在 Panel2 中）



图 6-8 案例界面设计效果

(3) 双击 Button 控件，并在添加如下代码：

```
protected void Button1_Click(object sender, EventArgs e)
{
    string strURL;
    strURL=
"TargetPage.aspx?userName="+txtName.Text+"&Pwd="+txtPwd.Text;
    Response.Redirect(strURL);
}
```

(4) 在 TargetPage.aspx 页面的 Page_Load 添加如下代码：

```
protected void Page_Load(object sender, EventArgs e)
{
    string UserNm;
    UserNm= Request.QueryString["userName"];
```

```

        Response.Write("欢迎" + UserNm + "<br>" );
        Response.Write( "您的密码是：" + Request.QueryString["Pwd"]);
    }

```

6.1.5 测试

运行程序结果如下图 6-9 所示,输入相应的信息后点击” 登录” 按钮,即可出现图 6-10 所示的效果。



图 6-9 程序运行效果

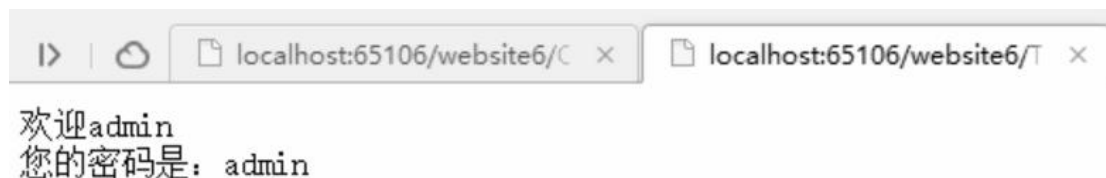


图 6-10 程序运行效果

【任务拓展】

案例：利用 Request 对象获取客户端浏览器能力信息

通过访问 Request 对象的 Browser 数据集合,可以容易地查询浏览器的能力。例如,下面的代码可以获取并输出浏览器描述、版本、是否支持 Cookie、DOM 版本号、是否安装 CLR、客户端操作系统等信息:

- (1) 在【website6】网站添加一个名为【Browser.aspx】的页面,
- (2) 在页面的 Page_load 事件中添加如下代码:

```

protected void Page_Load(object sender, EventArgs e)
{
    Response.Write("浏览器:" + Request.Browser.Browser);
    Response.Write("<br>版本:" + Request.Browser.Version);
    Response.Write("<br>支持 Cookie:" + Request.Browser.Cookies);
    Response.Write("<br> 微 软      D O M      版 本 号      : " +
Request.Browser.MSDomVersion.ToString());
    Response.Write("<br>W3C      D O M      版 本 号      : " +
Request.Browser.W3CDomVersion.ToString());
    Response.Write("<br>安装 CLR:" + Request.Browser.ClrVersion.ToString());
}

```

```

Response.Write("<br>客户端操作系统:" + Request.Browser.Platform);
}

```

运行结果如图 6-11 所示:



图 6-11 案例运行效果

6.2 子任务二：统计在线人数

【任务陈述】

同一个网站可以由许多用户访问，可以用 Application 对象记录整个网站的信息，实现统计在线人数的功能。

【知识准备】

6.2.1 Application 对象概述

Application 对象是公共对象，主要用于在所有用户间共享信息，所有用户都可以访问该对象中的信息并对信息进行修改。该对象多用于创建网站计数器和聊天室。

定义一个 Application 对象的语法格式如下：

```
Application[“属性名”]=值;
```

例如：

```
Application["online"] = 0;//在线人数
```

表 6-6 Application 对象的常用属性和方法

名称	属性/方法	说明
Count	属性	获取 HttpApplicationState 集合中的对象的个数
Item	属性	获取对 HttpApplicationState 集合中对象的访问
Add	方法	将新对象添加到 HttpApplicationState 集合中
Remove	方法	从 HttpApplicationState 集合中移除命名对象
Clear	方法	从 HttpApplicationState 集合中移除所有对象
GetKey	方法	通过索引获取 HttpApplicationState 的对象名
Lock	方法	锁定对 HttpApplicationState 变量的访问以促进访问同步
UnLock	方法	解锁对 HttpApplicationState 变量的访问以促进访问同步

其中 Lock 和 UnLock 方法是比较常用的两个方法，用于处理多个用户同时向 Applicationd 对象写入数据时可能存在的写入数据不一致的问题。先用 Lock 方法将 Application 对象锁定，阻止其他用户修改，确保同一时刻只有一个用户对该对象的信息进行修改。当用户完成修改信息后，在使用 UnLock 方法对 Application 对象进行解锁，下一个用户才能对 Application 对象进行修改。例如：在线人数统计使用 Lock 方法和 UnLock 方法实现锁定和解锁。

```
Application.Lock();
Application["online"] = int.Parse(Application["online"].ToString()) + 1;
Application.Unlock();
```

2. Application 对象的事件

Application 对象有以下 4 个事件。

(1) OnStart 事件：在整个 ASP.NET 应用程序启动时首先被触发的事件，即在一个虚拟目录中第一个 ASP.NET 程序执行时触发，该事件在程序的生命周期中仅被触发一次。

(2) OnEnd 事件：与 OnStart 事件正好相反，在整个应用停止时被触发，通常发生在服务器被重启/关机时，或 IIS 被停止时，该事件在程序的生命周期中仅被触发一次。

(3) OnBeginRequest 事件：在每一个 ASP.NET 程序被请求时就发生，即客户每访问一个 ASP.NET 程序时，就触发一次该事件。

(4) OnEndRequest 事件：ASP.NET 程序结束时，触发该事件。

这些事件属于全局事件，放在一个固定的文件 Global.asax 中，添加的方法如下：在网站的【解决方案资源管理器】中右击网站，选择【添加新项】，在弹出的对话框中选择【全局应用程序类】，可以看到此文件的默认名称为“Global.asax”，点击【添加】按钮即可。打开 Global.asax 可以看到并编写全局事件。

【实施与测试】

6.2.2 统计在线人数功能实现

(1) 修改 Global.asax 文件的 Application_Start 和 Application_End 事件

```
void Application_Start(object sender, EventArgs e)
{
    //在应用程序启动时运行的代码
    Application.Lock();
    Application["online"] = 0; //在线人数
    Application.Unlock();
}

void Application_End(object sender, EventArgs e)
{
    //在应用程序关闭时运行的代码
    Application.RemoveAll();
}
```

(2) 在上个任务建立的【ChatRoom.aspx.cs】页面进行修改，双击 Button 控件，修改为如下代码：

```
protected void Button1_Click(object sender, EventArgs e)
{
    Panel1.Visible = false;
    Panel2.Visible = true;
    Label1.Text = txtName.Text + “，您好！” + “<br>”;
    Application.Lock();
    Application["online"] =
int.Parse(Application["online"].ToString()) + 1; //在线人数加 1
    Application.Unlock();
    Label1.Text = Label1.Text + “当前在线人数为：” +
Application["online"].ToString() + “人”;
```

```
}
```

(4) 在 ChatRoom.aspx 页面的 Page_Load 添加如下代码:

```
protected void Page_Load(object sender, EventArgs e)
{
    Panel2.Visible = false;
}
```

6.2.3 测试

运行程序出现如图 6-12 所示界面, 输入相应的信息后点击“登录”按钮, 即可出现图 6-13 所示的效果。



图 6-12 程序运行效果



图 6-13 程序运行效果

【任务拓展】

实现图书馆管理系统网站的访问人数统计功能。

6.3 子任务三 在线留言

【任务陈述】

主要利用 Application 对象实现在在线留言功能, 游客可以查看聊天内容, 用户要输入用户名和密码登录后才能进行留言, 未登录进行留言者将被强制登录。

【知识准备】

6.3.1 Session 对象概述

Session 对象的功能和 Application 对象的功能类似, 都是用来存储跨网页程序的变量和对象, 但与 Application 对象不同的是, Session 对象为某一用户私有。这意味着, 在线的用户 A 不能访问同时在线的用户 B 的 Session 对象。

Session 对象也有其生存期。在默认的情况下, 如果浏览器在 20 分钟内没有再访问网站中的任何网页, 则该网站为其建立的 Session 对象将自动释放。而 Application 对象的生命周期终止于停止 IIS 服务。

设置 session 变量的方法: Session[“属性名”]=值

1. Session 常用的属性和方法

表 6-6 Session 常用的属性和方法

名 称	方 法 / 属性	描 述
Count	属性	获取 Contents 集合中的变量数
Item({name, index})	属性	从 Contents 集合内获取名称为 name 或下标为 index 的变量值
Timeout	属性	获取或设置 Session 对象的失效时间, 单位为分钟, 默认为 20 分钟
Clear	方法	清除 Contents 集合中的所有变量
Remove(name)	方法	从 Contentes 集合中删除名为 name 的变量
RemoveAll	方法	清除 Contents 集合中的所有变量
Add(name, value)	方法	向 Contents 集合中添加名为 name、值为 value 的变量

2. Session 对象事件

Session 对象有以下两个事件:

(1) OnStart 事件: 当用户第一次访问 ASP.NET 应用程序时将创建 Session 对象, 并触发 OnStart 事件。对同一用户该事件只发生一次, 除非发生 OnEnd 事件, 否则不会再触发该事件。

(2) OnEnd 事件: 在 Timeout 属性所设置的时间内没有再访问网页, 或者调用了 Abandon 方法都会触发此事件。该事件通常用于用户会话结束的处理, 如将数据写入文件或数据库等。注意, 仅当会话状态 mode 被设置为 InProc 时, 才会引发 OnEnd 事件。

Session 对象的生命周期始于用户第一次连接到应用程序任何网页, 在以下情况之一发生时结束:

(1) 断开与服务器的连接。

(2) 浏览者在 Timeout 属性规定的时间内未与服务器联系。

案例: 利用 Session 对象准确统计在线人数。

在子任务 6.2 中, 无论用户是首次访问还是刷新网页, 计数器都会加 1, 这样就产生了重复计数, 可以利用 Session 对象来区分是首次访问还是刷新网页, 避免重复统计在线人数, 打开 ChatRoom.aspx 页面, 修改 Button1_Click 部分代码如下:

```
if (Session.IsNewSession)
{
    Application.Lock();
    Application["online"]=int.Parse(Application["online"].ToString()) + 1;
    Application.UnLock();
}
```

案例: 利用 Session 强制登录

很多网站是有访问限制的, 有些网页只允许已登录的用户访问, 为了防止未登录用户跳过登录页而直接访问受保护的页面, 可以利用 Session 对象来判断用户的登录状态, 若未登录则将其转到登录状态, 从而起到了保护资源的目的。

例如: 未登录不能查看聊天内容代码如下:

```
string user="";
if (Session["username"] == "")
```

```

        Response.Redirect("login.aspx");
    else
        user = Session["username"].ToString();

```

3. Timeout 属性

很多网站中，用户登录之后，如果一段时间内不进行任何操作，则用户就会处于“超时”状态，此时，如果用户进行操作，则网站会跳转到登录界面，需要用户重新登录，该功能就是通过设置 Session 对象的会话时间来实现的。

案例：将 Session 对象的 Timeout 属性设置为 1 分钟，超时之后按“刷新”按钮，程序会自动跳转到聊天登录页面。

```
Session.Timeout = 1;
```

注意：Session 对象的 Timeout 属性是以分钟为单位，不超过 626600 分钟（1 年）的值，可以在应用程序的 Web.Config 文件中，使用 sessionState 配置元素的 timeout 属性来设置 Timeout 属性，例如，在 Web.Config 文件中设置 Session 对象的有效时间为 30 分钟，代码如下：

```
<sessionState mode="InProc" timeout="30" >
```

6.3.2 Cookie 对象

Cookie 对象是 Web 服务器保存在用户硬盘上的一段文本。Cookie 允许一个 Web 站点在用户的计算机上保存信息并且随后再取回它。信息的片段以“键/值”对的形式存储。

Cookie 对象用于保存客户端的访问信息。当用户第一次访问一个网站时，网站发送给用户的除了请求页面外，还有一个包含访问日期和时间的 Cookie。用户接收网页的同时，将服务器发来的 Cookie 保存在客户端硬盘上的某个文件夹中。以后用户再访问这个网站时，服务器会自动去硬盘上查找与之相关联的 Cookie，并将该 Cookie 与访问请求一起发送到站点。

Cookie 存储的数据量受限制，大多数浏览器支持的最大容量为 4096 字节，

注意：虽然 Application、Session 与 Cookie 都用于保存信息，但前两者将信息保存到服务器的内存中，后者将信息存放在客户端的硬盘上。也就是说，无论何时用户连接到服务器，Web 站点都可以访问 Cookie 信息，既方便用户的使用，由方便网站对用户的管理。

1. Cookie 变量的存取

要存储一个 Cookie 变量，其语法格式为：

```
Response.Cookie["变量名"].Value=值;
```

例如：

```
Response.Cookie["username"].Value="admin";//创建 cookie 变量保存用户名
```

如果要取回 Cookie，使用 Request 对象的 Cookie 集合，将指定的 Cookie 返回，语法格式为：

```
变量=Request.Cookies["变量名"].Value
```

例如：

```
Request.Write(Request.Cookies["username"].Value);//输出用户名
```

1. Cookie 的常用属性和方法

表 6-7 Cookie 的常用属性和方法

名称	属性/方法	描述
Name	属性	获取或设置 Cookie 变量的名称
Value	属性	获取或设置 Cookie 变量的数值
Items	属性	按对象的索引编号或名称获取 Cookie 集合中的对象

Expires	属性	获取或设置 Cookie 的过期时间，应为 DateTime 类型的数据
Add	方法	新增一个 Cookie 变量
Clear	方法	清除 Cookie 集合中的变量
Get	方法	通过变量名或索引得到 cookie 的变量值
GetKey	方法	以索引值来获取 Cookie 的变量名称
Remove	方法	通过 Cookie 变量名来删除 Cookie 变量

案例：运用 Cookie 实现密码记忆功能。当用户第一次登录输入用户名和密码，并选中“记住密码”复选框后，程序会将用户的用户名和密码存储到 Cookie 中。当下次登录时，当用户输入用户名后，程序会查找 Cookie 中是否存在该用户名，并获取相应的密码。案例运行结果如图 6-14 所示。

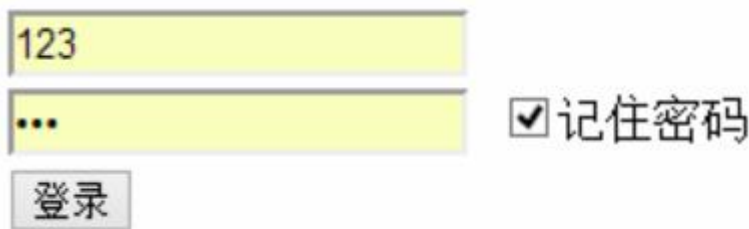


图 6-14 程序运行效果

案例代码如下：

```
protected void Page_Load(object sender, EventArgs e)
{
    string username = this.txtUserName.Text;//用户名
    string password = this.txtPassword.Text;//密码
    if (CheckBox1.Checked)
    {
        //判断客户端浏览器是否存在该 Cookie 存在就先清除
        if (Request.Cookies["username"] != null &&
Request.Cookies["password"] != null)
        {
            Response.Cookies["username"].Expires =
System.DateTime.Now.AddSeconds(-1);//Expires 过期时间
            Response.Cookies["password"].Expires =
System.DateTime.Now.AddSeconds(-1);
        }
        else
        {
            //向客户端浏览器加入 Cookie （用户名和密码 最好是使用 MD6 加密）
            HttpCookie hcUserName1 = new HttpCookie("username");
            hcUserName1.Expires = System.DateTime.Now.AddDays(7);
            hcUserName1.Value = username;
            HttpCookie hcPassword1 = new HttpCookie("password");
            hcPassword1.Expires = System.DateTime.Now.AddDays(7);
            hcPassword1.Value = password;
        }
    }
}
```

```

        Response.Cookies.Add(hcUserName1);
        Response.Cookies.Add(hcPassword1);
    }
}
protected void Button1_Click(object sender, EventArgs e)
{
}
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    if (Request.Cookies["username"] != null)
    {
        if
(Request.Cookies["username"].Value.Equals(txtUserName.Text.Trim()))
        {
            txtPassword.Attributes["value"] =
Request.Cookies["username"].Value;
        }
    }
}

```

【实施与测试】

6.3.3 在线留言功能实现

(1) 完善界面设计, 在【ChatRoom.aspx】中添加下表所示控件并设置相应的属性。

表 6-8 控件及其属性设置

功能	控件	属性
留言显示	TextBox	ID="txtChat" TextMode="MultiLine"
留言发表	TextBox	ID="txtSend"
提交	Button	ID="btnSubmitChat "
登出	Button	登出

(2) 修改 Glocal.aspx 文件的 Application_Start、Session_Start 和 Session_End 事件

```

void Application_Start(object sender, EventArgs e)
{
    //在应用程序启动时运行的代码
    Application.Lock();
    Application["online"] = 0; //在线人数
    Application["content"] = ""; //聊天内容
    Application.Unlock();
}

void Session_Start(object sender, EventArgs e)
{
    ////在新会话启动时运行的代码
    Session["username"] = ""; //建立一个 session, 用于保存用户名
}

```

```

        Session.Timeout = 1;
        Application.Lock();
        Application["online"] = int.Parse(Application["online"].ToString()) +
1; //在线人数加 1
        Application.Unlock();
    }
    void Session_End(object sender, EventArgs e)
    {
        Application.Lock();
        Application["online"] = int.Parse(Application["online"].ToString()) -
1; //在线人数减 1
        Application.Unlock();
    }

```

(3) 在上个任务建立的【ChatRoom.aspx.cs】页面进行修改，修改后的代码如下：

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["username"].ToString() == "")
        Panel2.Visible = false;
        txtChat.Text = Application["content"].ToString();
}
protected void btnSubmit_Click(object sender, EventArgs e)
{
    Panel1.Visible = false;
    Panel2.Visible = true;
    Label1.Text = txtName.Text + ", 您好!" + "<br>";
    if (Session.IsNewSession)
    {
        Application.Lock();
        Application["online"] = int.Parse(Application["online"].ToString())
+ 1; //在线人数加 1
        Application.Unlock();
    }
    Label1.Text = Label1.Text + " 当前在线人数为: " +
Application["online"].ToString() + "人";
    Session["username"] = txtName.Text;
    string username = this.txtName.Text; //用户名
    string password = this.txtPwd.Text; //密码

    if (ckbPwd.Checked)
    {
        //判断客户端浏览器是否存在该 Cookie 存在就先清除
        if (Request.Cookies["username"] != null &&
Request.Cookies["password"] != null)
        {

```

```

        Response.Cookies["username"].Expires =
System.DateTime.Now.AddSeconds(-1); //Expires 过期时间
        Response.Cookies["password"].Expires =
System.DateTime.Now.AddSeconds(-1);
    }
    else
    {
        //向客户端浏览器加入 Cookie （用户名和密码 最好是使用 MD6 加密）
        HttpCookie hcUserName1 = new HttpCookie("username");
        hcUserName1.Expires = System.DateTime.Now.AddDays(7);
        hcUserName1.Value = username;
        HttpCookie hcPassword1 = new HttpCookie("password");
        hcPassword1.Expires = System.DateTime.Now.AddDays(7);
        hcPassword1.Value = password;
        Response.Cookies.Add(hcUserName1);
        Response.Cookies.Add(hcPassword1);
    }
}

protected void Button2_Click(object sender, EventArgs e)
{
    if (Session["username"].ToString() != "")
    {
        Session["username"] = "";
        Response.Redirect("ChatRoom.aspx");
    }
}

protected void btnSumbitChat_Click(object sender, EventArgs e)
{
    string user="";
    if (Session["username"] == "")
        Response.Write("ChatRoom.aspx");
    else
    {
        user = Session["username"].ToString();
        string msg = "\n" + user + "说:" + txtSend.Text.ToString() + "\n"; ;
        Application.Lock();
        Application["content"] = (string)Application["content"] + msg;
        Application.Unlock();
        txtChat.Text = (string)Application["content"];
    }
}

protected void txtName_TextChanged(object sender, EventArgs e)

```

```

{
    if (Request.Cookies["username"] != null)
    {
        if (Request.Cookies["username"].Value.Equals(txtName.Text.Trim()))
        {
            txtPwd.Attributes["value"] = Request.Cookies["username"].Value;
        }
    }
}

```

(4) 【ChatRoom.aspx】生成的代码如下:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="ChatRoom.aspx.cs" Inherits="Default2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:Panel ID="Panel1" runat="server">
        请 您 登 陆 ! 姓 名 : <asp:TextBox ID="txtName" runat="server"
AutoPostBack="True"
        ontextchanged="txtName_TextChanged"></asp:TextBox>
        密 码 : <asp:TextBox ID="txtPwd" runat="server"
TextMode="Password"></asp:TextBox>
        <asp:CheckBox ID="ckbPwd" runat="server" Checked="True" Text="记住密码"
AutoPostBack="True" />
        <asp:Button ID="btnSubmit" runat="server" onclick="btnSubmit_Click"
Text="登录" style="width: 60px" /> </asp:Panel>
    <asp:Panel ID="Panel2" runat="server">
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
    </asp:Panel>
    <asp:TextBox ID="txtChat" runat="server" Height="100px" Width="826px"
        ontextchanged="txtChat_TextChanged" TextMode="MultiLine"></asp:TextBox>
    <br />
    请 您 发 言 : <asp:TextBox ID="txtSend" runat="server" Height="63px"
Width="424px"></asp:TextBox>
    <asp:Button ID="btnSumbitChat" runat="server" onclick="btnSumbitChat_Click"
Text="提交" />
        <asp:Button ID="Button2" runat="server" onclick="Button2_Click"
Text="退出" />
</asp:Content>

```

(5)运行结果如下:



图 6-15 程序运行效果

(6)输入姓名和密码，选中“记住密码”复选框，点击“登录按钮”后出现如图 6-16 所示界面。显示了当前的在线人数。



图 6-16 程序运行效果

(7)在留言处输入想要发表的话，点击”按钮”出现如图 6-17 所示界面。



图 6-17 程序运行效果

【任务拓展】

查阅资料，利用 Cookie 实现“您最喜欢的书籍”在线投票功能。



项目重现

完成网上购物系统的留言板

1. 项目目标

完成本项目后，您能够：

- 完成网上购物系统的留言板
- 统计网上购物系统的在线人数、网站访问量
- 网上购物系统用户登录的密码记忆功能以及强制登录功能

2. 知识目标

完成本项目后，您应该掌握

- 用 Response 和 Request 对象进行页面传值
- 用 Request 对象获取客户端浏览器的信息
- 用 Application 对象统计在线人数、实现消息发送
- 用 Session 对象精确统计在线人数、实现强制登录
- 用 Cookie 实现密码记忆功能

3. 项目介绍

网上购物系统通常具有在线留言，登录密码记忆，统计在线人数、网站访问量以及对具有访问权限的页面进行强制登陆功能。

4. 项目内容

- (1) 在网上购物系统各页面之间进行传值
- (2) 为用户显示客户端浏览器的信息
- (3) 统计在线人数，实现在线聊天，强制登陆
- (4) 实现网上购物系统登录时密码记忆功能