

ASP.NET动态网页设计

执教：朱亚林



南通师范高等专科学校
Nantong Normal College



ASP.NET内置对象

ASP.NET 内置对象

- Response对象
- Request对象
- Application对象
- Session对象
- Cookie对象
- Server对象

这些对象使用户更容易收集通过浏览器请求发送的信息、响应浏览器以及存储用户信息，以实现其他特定的状态管理和页面信息的传递。

Session对象

Session对象



- Session中文是“会话”的意思，在ASP.NET中代表了服务器与客户端之间的“会话”。
- Session的作用时间从用户到达某个特定的Web页开始，到该用户离开Web站点，或在程序中利用代码终止某个Session结束。
- 引用Session 则可以让一个用户访问多个页面之间的切换也会保留该用户的信息。
- 系统为每个访问者都设立一个独立的Session对象，用以存储Session变量，并且各个访问者的Session对象互不干扰。

Session常用属性和方法

名称	类型	说明
TimeOut	属性	会加或设定Session对象变量的有效时间。当使用者超过有效时间而没有动作时，Session对象就会失效。默认时间20分钟。
Abandon	方法	此方法用于结束当前会话，并清除会话中的所有信息。
Clear	方法	此方法用于清除全部的Session对象变量，但不结束会话。

实例：用户登录认证

思路：

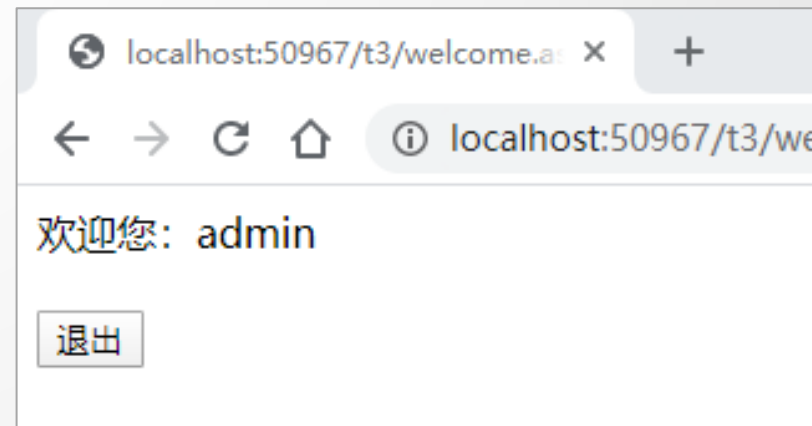
- (1) 提供一个入口给用户输入用户名和密码
- (2) 将输入的内容与正确的信息进行比对，如果完全一致，则为其建立一个Session对象。
- (3) 在后续需要登录访问的页面中，则可以检测该Session对象是否存在，如在则可以访问，如不在则拒绝访问。



“学籍管理系统”后台管理页面

用户名

密码



实例：用户登录认证

■ 关键代码

Login.aspx

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (TextBox1.Text == "admin" && TextBox2.Text == "admin")
    {
        Session["user"] = TextBox1.Text;
        Response.Redirect("~/welcome.aspx");
    }
    else
    {
        Response.Write("<script>alert(' 请检查用户名或密码是否正确! ')</script>");
    }
}
protected void Button2_Click(object sender, EventArgs e)
{
    TextBox1.Text = "";
    TextBox2.Text = "";
}
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["user"] != null)
    {
        Response.Write("欢迎您: " + Session["user"]);
    }
    else
    {
        Response.Write("请登录! ");
        Response.Write("<meta http-equiv='Refresh' content='3;URL=login.aspx' /> ");
    }
}
```

welcome.aspx

Cookie对象

Cookie对象

- **Cookie**对象用于保存客户端浏览器请求的服务器页面，也可用于存放非敏感性的用户信息，信息保存的时间可以根据用户的需要进行设置。
- 并非所有的浏览器都支持**Cookie**，并且数据信息是以文本的形式保存在客户端计算机上的。

Cookie常用属性和方法

名称	类型	说明
Expires	属性	设定Cookie变量的有效时间，默认为1000分钟。若设为0，则可以实时删除Cookie变量
Name	属性	取得Cookie变量的名称
Value	属性	获取或设置Cookie变量的内容值
Path	属性	获取或设置Cookie适用的URL
Equals	方法	确定指定Cookie是否等于当前的Cookie
ToString	方法	返回此Cookie对象的一个字符串表示形式

Expires属性的赋值

■ Expires, 设置Cookie变量的有效时间。

例:

```
DateTime.Now.AddMonths(1); //一个月后过期
```

```
DateTime.Parse(“10/26/2019”);
```

```
DateTime.MaxValue; //永不过期
```

```
DateTime.MinValue; //关闭浏览器后过期
```

Cookie对象的设置和读取

设置:

```
Response.Cookies[varName].Value = "something";
```

读取:

```
SOMETHING = Request.Cookies[varName].Value;
```

实例：用户免密码登录

- 用户登录成功后，再次打开页面时会自动读取已保存的Cookie信息并进行登录验证。

```
if (TextBox1.Text == "admin" && TextBox2.Text == "admin")
{
    Session["user"] = TextBox1.Text;

    Response.Cookies["user"].Value=Session["user"].ToString();
    Response.Cookies["passwd"].Value = TextBox2.Text;
    Response.Cookies["user"].Expires = DateTime.MaxValue;
    Response.Cookies["passwd"].Expires = DateTime.MaxValue;
    Response.Cookies["passwd"].Value = TextBox2.Text;
    Response.Redirect("~/welcome.aspx");
}
```

设置Cookie

```
protected void Button2_Click(object sender, EventArgs e)
{
    for (int i = 0; i < Request.Cookies.Count; i++)
    {
        Response.Cookies[Request.Cookies[i].Name].Expires = DateTime.Now.AddDays(-1);
    }
}
```

清除Cookie

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Cookies["user"] != null && Request.Cookies["passwd"] != null)
    {
        if (Request.Cookies["user"].Value == "admin" && Request.Cookies["passwd"].Value == "admin")
        {
            Session["user"] = Request.Cookies["user"].Value;
            Response.Redirect("~/welcome.aspx");
        }
    }
}
```

加载Cookie

Server对象

Server对象

- **Server**对象定义了一个与**Web**服务器相关的类，提供对服务器上的方法和属性的访问，用于访问服务器上的资源。

Server对象

■ 常用的属性有：

MachineName，获取服务器的计算机名称

ScriptTimeout，获取和设置请求超时值

Server对象

常用的方法有：

Execute，在当前请求的上下文中执行指定资源的处理程序，然后将控制返回给该处理程序。（执行完毕还回到原页面）

Transfer，终止当前页面的执行，并为当前请求开始执行新页。（执行到哪里，就到哪里）

MapPath，返回与**Web**服务器上的指定虚拟路径相对应的物理文件路径。

HtmlDecode，对已被编码以消除无效**HTML**字符的字符串进行解码。

HtmlEncode，对要在浏览器中显示的字符串进行编码。

UrlEncode，编码字符串，以便通过**URL**从服务器到客户端进行可靠的**HTTP**传输。

UrlDecode，解码字符串，该字符串为了进行**HTTP**传输而进行编码并在**URL**中发送到服务器。

思考：对比Execute和Transfer的异同

```
protected void Button1_Click(object sender, EventArgs e)
{
    Server.Execute("hello.aspx?name=Jack");
    Response.Write(Session["user"].ToString());
}
protected void Button2_Click(object sender, EventArgs e)
{
    Server.Transfer("hello.aspx?name=Jack");
    Response.Write(Session["user"].ToString());
}
```



实例

- 使用MapPath获取服务器和物理地址，以及显示服务器的计算机名

```
protected void Button3_Click(object sender, EventArgs e)
{
    Response.Write(Server.MapPath("."));
    Response.Write(Server.MachineName);
}
```