本章教学目标

- 内置对象的基本概念
- 内置对象的相关属性和方法
- 内置对象的使用

ASP．NET提供了内置对象有Request、Response、Application、Session和Server等。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。

5-1 Request 对象

Request对象主要是让服务器取得客户端浏览器的一些数据,包括从HTML表单用Post或者GET方法传递的参数、Cookie和用户认证。因为Request对象是Page对象的成员之一，所以在程序中不需要做任何的声明即可直接使用，其类名为 HttpRequest。

主要的**属性**有：

1．Form属性

通过该属性，获取以Post方式提交的表单数据。

5-1 Request 对象

2．QueryString属性

通过该属性可以获取HTTP查询字符串变量集合或以Get方式提交的表单数据。例如下面是带有查询字符串变量的链接地址：

http://localhost/index.aspx?uname=tom&pwd=abc

在index.aspx.cs文件中编写page_load事件的处理程序，获取查询字符串变量的内容：

```
void Page_Load(object sender, System.EventArgs e)
{
 Response.Write("变量uname的值：" + Rec
+"<br>");
Response.Write("变量pwd的值：" + Reque
}
```

程序的输出为：
变量uname的值：tom
变量pwd的值：abc

5-1 Request 对象

3．UserHostAddress属性

通过该属性可以获获取远程客户端的IP主机地址。

4．Browser属性

通过该属性可以获获取有关正在请求的客户端的浏览器功能的信息。

主要的方法有：

1．BinaryRead()方法

执行对当前输入流进行指定字节数的二进制读取。

5-2 Response 对象

Response对象可以输出信息到客户端，包括直接发送信息给浏览器、重定向浏览器到另一个URL或设置cookie的值，Response对象同样可以直接使用，其类名为HttpResponse。

主要的方法有：

1．Write()方法

将指定的字符串或表达式的结果写到当前的HTTP输出。例如下面的代码在网页中输出了100个数字。

```
for(int i=1;i<=100;i++)
{
    Response.Write("i= "+i+"<BR>");
}
```

5-2 Response 对象

2．End()方法

停止页面的执行并得到相应结果。例如：

```
int N=100;
Response.Write("N=" + N + "<br>");
Response.End();
Response.Write("该值的平方值是：" +
N*N);
```

程序的输出为：
N=100

5-2 Response 对象

3．Clear()方法

用来在不将缓存中的内容输出的前提下，清空当前页的缓存，仅当使用了缓存输出时，才可以利用Clear方法。

4．Flush()方法

将缓存中的内容立即显示出来。该方法有一点和Clear方法一样，它在脚本前面没有将Buffer属性设置为True时会出错。和End方法不同的是，该方法调用后，该页面可继续执行。

5．Redirect()方法

使浏览器立即重定向到程序指定的URL。例如执行下面的代码，浏览器会跳转到网易的主页：

```
Response.Redirect("http://www.163.com");
```

5-3 Application对象

Application对象是HttpApplicationState类的单个实例，将在客户端第一次从某个特定的ASP.NET应用程序虚拟目录中请求任何URL资源时创建。对于Web服务器上的每个ASP.NET应用程序，都要创建一个单独的实例。然后通过内部Application对象公开对每个实例的引用。

Application对象使给定应用程序的所有用户之间共享信息，并且在服务器运行期间持久地保存数据。因为多个用户可以共享一个Application对象，所以必须要有Lock和Unlock方法，以确保多个用户无法同时改变某一属性。Application对象成员的生命周期止于关闭IIS或使用Clear方法清除。

5-3 Application对象

主要的**属性**有：

1．AllKeys属性
获取HttpApplicationState集合中的访问键。

2．Count属性
获取HttpApplicationState集合中的对象数。

主要的方法有：

1．Add()方法
新增一个新的Application对象变量。

2．Clear()方法
清除全部的Application对象变量。

5-3 Application对象

3．Get()方法
使用索引关键字或变量名称得到变量值。

4．GetKey()方法
使用索引关键字获取变量名称。

5．Lock()方法
锁定全部的Application变量。

6．Remove()方法
使用变量名称删除一个Application对象。

7．RemoveAll()方法
删除全部的Application对象变量。

8．Set()方法
使用变量名更新一个Application对象变量的内容。

9．UnLock()方法
解除锁定的Application变量。

5-3 Application对象

Application对象的使用代码如下：

```
Application.Add("App1","Value1");//或Application[“App1”] = Value1;
Application.Add("App2","Value2");
Application.Add("App3","Value3");
int N;
for(N=0;N<Application.Count;N++)
{
        Response.Write("变量名："+ Application.GetKey(N));//获取变量名称
        Response.Write("变量值："+ Application.Get(N) +"<br>");//获取变量值，也可以写成Application[N],Appllicatio
}
Application.Clear();//清除Application中的变
```

代码的输出为：
变量名：App1变量值：Value1
变量名：App2变量值：Value2
变量名：App3变量值：Value3

5-3 Application对象

Lock方法可以阻止其他客户修改存储在Application对象中的变量，以确保在同一时刻仅有一个客户可修改和存取Application变量。如果用户没有明确调用Unlock方法，则服务器将在页面文件结束或超时即可解除对Application对象的锁定。

方法可以使其他客户端在使用Lock方法锁住Application对象后，修改存储在该对象中的变量。如果未显式地调用该方法，Web服务器将在页面文件结束或超时后解锁Application对象。代码编写如下所示：

```
Application.Lock();
Application["变量名"]="变量值";
Application.UnLock();
```

5-4 Session对象

Session对象是HttpSessionState的一个实例。该类为当前用户会话提供信息，还提供对可用于存储信息的会话范围的缓存的访问，以及控制如何管理会话的方法。

可以使用Session对象存储特定用户会话所需的信息。这样，当用户在应用程序的Web页之间跳转时，存储在Session对象中的变量将不会丢失，而是在整个用户会话中一直存在下去。

当用户请求来自应用程序的Web页时，如果该用户还没有会话，则Web服务器将自动创建一个Session对象。当会话过期或被放弃后，服务器将中止该会话。

5-4 Session对象

主要的属性有:

1．Count属性

获取会话状态集合中Session对象的个数。

2．TimeOut属性

获取并设置在会话状态提供程序终止会话之前各请求之间所允许的超时期限，超时期限（以分钟为单位）。

3．SessionID属性

获取用于标识会话的唯一会话ID。

例如：

```
Session["Session1"]="Value1";//设置Session中的变量值
Session["Session2"]="Value2";
Session.Timeout=1;
string s1 =Session["Session1"].ToString();//获取Session中的变量值
string s2 =Session["Session2"].ToString();
```

5-4 Session对象

主要的方法有：

1．Add()方法
新增一个Session对象。

2．Clear()方法
清除会话状态中的所有值。

3．Remove()方法
删除会话状态集合中的项。

4．RemoveAll()方法
清除所有会话状态值。

例如：

```
int userId = 1;
string userName = "test";
string userPwd = "123456";
Session.Add("userId",userId);//增加一个对象
Session.Add("userName", userName);
Session.Add("userPwd", userPwd);
Session.RemoveAll();//清除所有对象
```

5-5 Server对象 ≥

Server对象是HttpServerUtility的一个实例。该对象提供对服务器上的方法和属性的访问。

主要的**属性**有：

1．MachineName属性
获取服务器的计算机名称。

2．ScriptTimeout属性
获取和设置请求超时（以秒计）。

主要的**方法**有：

1．Execute()方法
使用另一页执行当前请求。

5-5 Server对象

2．Transfer()方法

终止当前页的执行，并为当前请求开始执行新页。

Execute和Transfer方法的区别是：

Transfer的执行方式：第一个页面跳转到第二个页面时，页面处理的控制权也进行移交，但浏览器的Url仍保存第一个页面的URL信息。这种重定向请求在服务器端执行，客户端并不知道服务器执行页面跳转操作。

Execute的执行方式：允许当前页面执行同一web服务器的另一页面，当另一页面执行完毕后，控制流程重新返回到原页面。

5-5 Server对象

3．HtmlDecode()方法

对已被编码以消除无效HTML字符的字符串进行解码。

4．HtmlEncode()方法

对要在浏览器中显示的字符串进行编码。例如：

```
String  str= "<font color='red'><i>Server对象的使用</i></font>”;
Response.Write("字符串不经Html编码直接输出：<br>");
Response.Write(str);
Response.Write("<p>字符串经过Html编码后输出：<br>");
String strHtmlC
Response.Write
Response.Write
strHtmlContent
Response.Write
```

程序的输出为：
字符串不经Html编码直接输出：
Server对象的使用
字符串经过Html编码后输出：
 Server对象的使用
对编码后的字符串进行解码：
Server对象的使用

5-5 Server对象

5．MapPath()方法

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。例如，创建的网站目录为：D:\mysite,在网站内存在Default.aspx网页，通过MapPath()方法返回其物理文件路径。

```
string  FilePath = Server.MapPath("Defa
Response.Write(FilePath);
```

程序的输出为：
D:\mysite \Default.aspx

6．UrlDecode()方法

对字符串进行解码，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。

7．UrlEncode()方法

编码字符串，以便通过URL从Web服务器到客户端进行可靠的HTTP传输。

典型案例5-1：车辆基本信息查询系统

一、案例功能说明

本章典型案例，主要是实现一个车辆基本信息查询系统。将车辆的信息以对象的形式保存在Session对象中，通过用户输入的车牌号关键字，找到指定的车辆对象，并将车辆信息进行输出。

二、案例要求

（1）使用Session对象和对象数组保存车辆信息。

（2）使用Response的Redirect方法实现页面跳转。

（3）使用Request的QueryString属性获取用户输入的车牌号。

典型案例5-1：车辆基本信息查询系统

三、操作和实现步骤

（1）建立一个工程文件夹为：车辆基本信息查询系统。

（2）启动VS2010，选择新建网站，建立空网站。添加名为Default.aspx和Main.aspx的页面，并设计Default.aspx为用户输入车牌号，进行查询的界面。如下图所示。

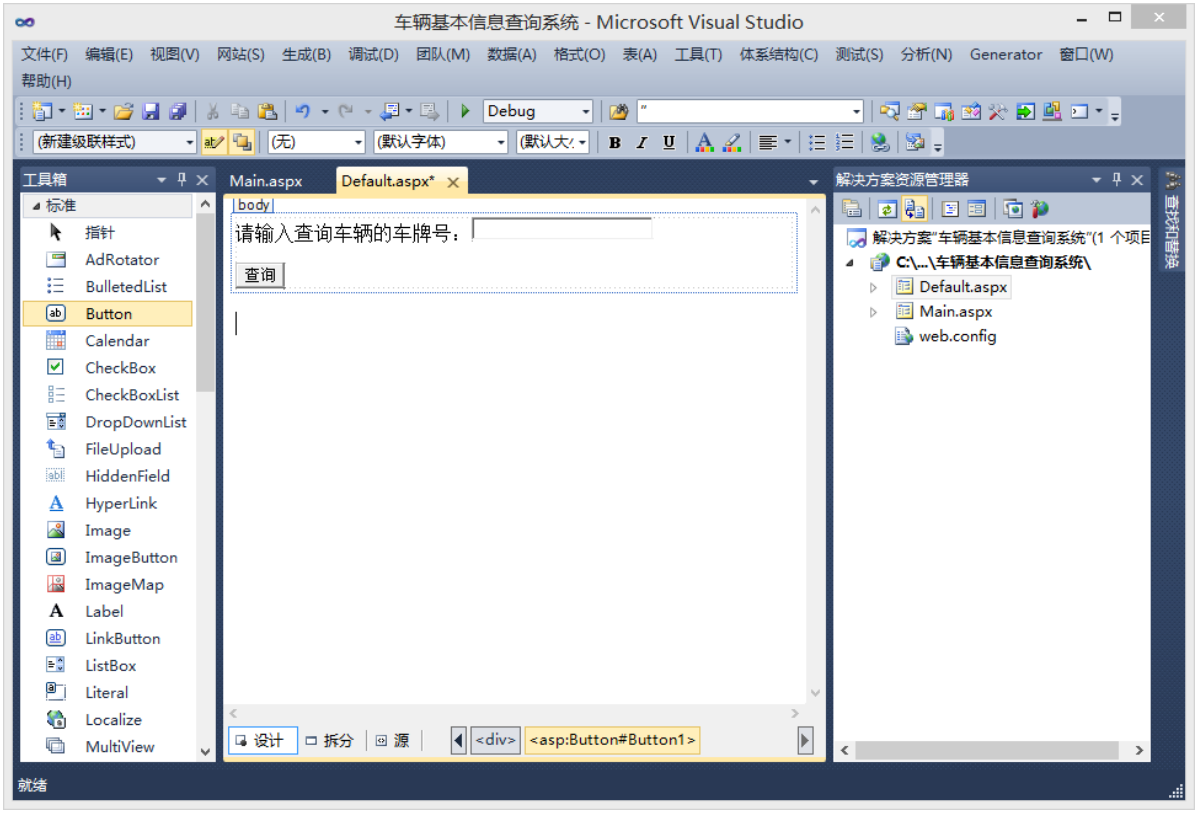

图5-1 设计查询界面