

第 3 章

ASP.NET 内置对象

本章的主要任务是学习 ASP.NET 内置对象的基本用法,重点是学习各个内置对象的基本功能,掌握其主要属性和方法。

由于各内置对象都有极其丰富的属性、方法,使用场合也各异,建议读者以本书示例为线索,重点掌握各对象的主要应用方法。

在 ASP 的开发中,这些内置对象已经存在,这些内置对象包括 Response、Request、Application 等,虽然 ASP 是一种可以称得上是“过时的”技术,但是 ASP.NET 开发人员依旧可以使用这些对象。这些对象不仅能够获取页面传递的参数,某些对象还可以保存用户的信息,如 Cookie、Session 等。

ASP.NET 提供了 7 个内置对象,如表 3-1 所示。这些对象可以在页面中直接使用,通过 ASP.NET 内置对象,在 ASP.NET 页面上以及页面之间可方便地实现获取、输出、传递、保留各种信息等操作,以完成各种复杂的功能。

表 3-1 ASP.NET 内置对象

对象名称	功能描述	对象名称	功能描述
Request	从浏览器获取信息	Session	用来保留客户端信息,保留在服务器端
Response	向浏览器输出信息	Server	获取服务器端信息
Application	为所有用户提供共享信息的手段	Trace	提供在 HTTP 页输出自定义跟踪和信息
Cookies	用来保留客户端信息,保留在客户端		

3.1 Request 对象概述

3.1.1 Request 对象常用属性和方法

当某浏览器向 Web 服务器请求一个 Web 页面时,Web 服务器就会收到一个 HTTP 请求,该请求包含用户、用户 PC、用户使用的浏览器等一系列信息。在 ASP.NET 中,可以通过 Request 对象设置或获取这些信息,Request 是 ASP.NET 最常用的对象之一。

该对象可以使用户获得 Web 请求的 HTTP 数据包的全部信息,其常用属性、方法及说明如表 3-2 和表 3-3 所示。

表 3-2 Request 对象常用属性及说明

属 性	说 明
ApplicationPath	获取服务器上 ASP.NET 虚拟应用程序的根目录路径
Browser	获取或设置有关正在请求的客户端浏览器的功能信息
ContentLength	指定客户端发送的内容长度(以字节计)
Cookies	获取客户端发送的 Cookie 集合
FilePath	获取当前请求的虚拟路径
Files	获取采用大部分 MIME 格式的由客户端上传的文件集合
Form	获取窗体变量集合
Item	从 Cookies、Form、QueryString 或 ServerVariables 集合中获取指定的对象
Params	获取 QueryString、Form、ServerVariables 和 Cookies 项的组合集合
Path	获取当前请求的虚拟路径
QueryString	获取 HTTP 查询字符串变量集合
UserHostAddress	获取远程客户端 IP 主机地址
UserHostName	获取远程客户端 DNS 名称

表 3-3 Request 对象常用方法及说明

方 法	说 明
MapPath	将当前请求的 URL 中的虚拟路径映射到服务器上的物理路径
SaveAs	将 HTTP 请求保存到磁盘

3.1.2 获取页面间传送的值

Request 方法通过 Params 属性和 QueryString 属性获取页面间的传值。

本例主要通过 Request 对象的不同属性实现获取请求页的值。执行程序,单击“跳转”按钮,运行结果如图 3-1 所示。实例位于 ASP.NET 内置对象中的 ch03。



图 3-1 输出二进制图像

程序实现的主要步骤如下。

(1) 新建一个网站,默认主页为 Default.aspx。在页面上添加一个 Button 控件, ID 属性设置为 btnRedirect, Text 属性设置为“跳转”。在按钮的 btnRedirect_Click 事件中实现页面跳转并传值的功能。代码如下:

```
protected void btnRedirect_Click(object sender, EventArgs e)
```

```
{  
    Response.Redirect("Request.aspx?value=获得页面间的传值");  
}
```

(2) 在该网站中添加一个新页,将其命名为“Request.aspx”。在页面 Request.aspx 的初始化事件中用不同方法获取 Response 对象传递过来的参数,并将其输出在页面上。代码如下:

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Response.Write("使用 Request[string key]方法"+Request["value"]+"<br>");  
    Response.Write("使用 Request.Params[string key]方法"+  
    Request.Params["value"]+"<br>");  
    Response.Write("使用 Request.QueryString[string  
    key]方法"+Request.QueryString["value"]+"<br>");  
}
```

3.1.3 获取客户端浏览器信息

用户通过使用 Request 对象的 Browser 属性访问 HttpBrowserCapabilities 属性可以获得当前正在使用的浏览器类型,并且可以获知该浏览器是否支持某些特定功能。下面就通过一个实例进行介绍。

本例主要通过 Request 对象的 Browser 属性获取客户端浏览器信息。执行程序,运行结果如图 3-2 所示。实例位于 ASP.NET 内置对象中的 ch04。

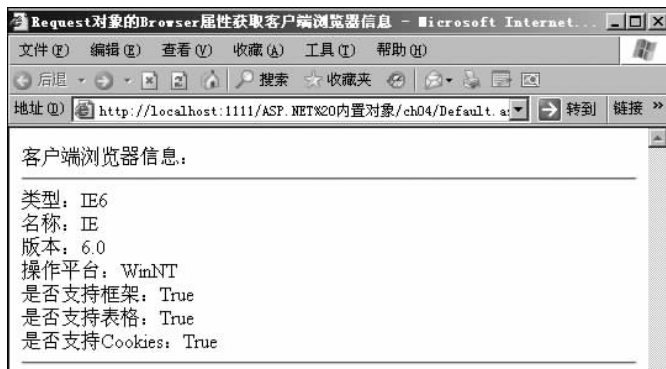


图 3-2 获取客户端浏览器信息

程序实现的主要方法如下。

新建一个网站,默认主页为 Default.aspx。在 Default.aspx 的 Page_Load 事件中先定义 HttpBrowserCapabilities 的类对象,用于获取 Request 对象的 Browser 属性的返回值。代码如下:

```
protected void Page_Load(object sender, EventArgs e)  
{  
    HttpBrowserCapabilities b=Request.Browser;  
    Response.Write("客户端浏览器信息: ");  
    Response.Write("<hr>");  
    Response.Write("类型: "+b.Type+"<br>");  
}
```

```

    Response.Write("名称: "+b.Browser+"<br>");
    Response.Write("版本: "+b.Version+"<br>");
    Response.Write("操作平台: "+b.Platform+"<br>");
    Response.Write("是否支持框架: "+b.Frames+"<br>");
    Response.Write("是否支持表格: "+b.Tables+"<br>");
    Response.Write("是否支持 Cookies: "+b.Cookies+"<br>");
    Response.Write("<hr>");
}

```

3.2 Response 对象的功能、常用属性、方法和示例

3.2.1 Response 对象概述

Response 对象用于将数据从服务器发送回浏览器。它允许将数据作为请求的结果发送到浏览器中,并提供有关响应的信息,可以用来在页面中输入数据、在页面中跳转,还可以传递各个页面的参数。它与 HTTP 协议的响应消息相对应。

3.2.2 Response 对象常用属性、方法

该对象将 HTTP 响应数据发送到客户端,并包含有关该响应的信息。其常用属性、方法及其说明如表 3-4 和表 3-5 所示。

表 3-4 Response 对象常用属性及说明

属 性	说 明
Buffer	获取或设置一个值,该值指示是否缓冲输出,并在完成处理整个响应之后将其发送
Cache	获取 Web 页的缓存策略,如过期时间、保密性、变化子句等
Charset	设定或获取 HTTP 的输出字符编码
Expires	获取或设置在浏览器上缓存的页过期之前的分钟数
Cookies	获取当前请求的 Cookie 集合
IsClientConnected	传回客户端是否仍然和 Server 连接
SuppressContent	设定是否将 HTTP 的内容发送至客户端浏览器,若为 True,则网页将不会传至客户端

表 3-5 Response 对象常用方法及说明

方 法	说 明
AddHeader	将一个 HTTP 头添加到输出流
AppendToLog	将自定义日志信息添加到 IIS 日志文件
Clear	将缓冲区的内容清除
End	将目前缓冲区中所有的内容发送至客户端后关闭
Flush	将缓冲区中所有的数据发送至客户端
Redirect	将网页重新导向另一个地址
Write	将数据输出到客户端
WriteFile	将指定的文件直接写入 HTTP 内容输出流

3.2.3 在页面中输出数据

Response 对象通过 Write 方法或 WriteFile 方法在页面上输出数据。输出的对象可以是字符、字符数组、字符串、对象或文件。

本例主要使用 Write 方法和 WriteFile 方法实现在页面上输出数据。在运行程序之前，在网站根目录下新建一个 TextFile.txt 文件，文件内容为“asp.net 编程词典”。执行程序，运行结果如图 3-3 所示。

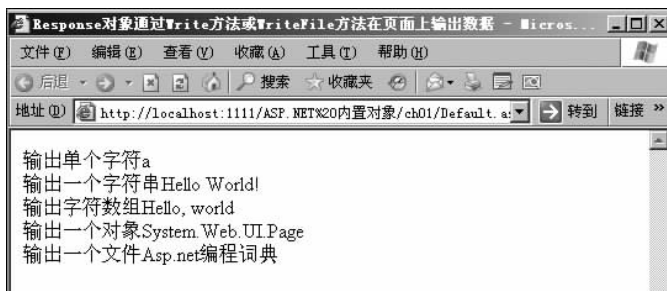


图 3-3 在页面中输出数据

程序实现的主要方法如下。

新建一个网站，默认主页为 Default.aspx。在 Default.aspx 的 Page_Load 事件中先定义 4 个变量，分别为字符型变量、字符串变量、字符数组变量和 Page 对象，然后将定义的数据在页面上输出。代码如下：

```
protected void Page_Load(object sender, EventArgs e)
{
    char c='a'; //定义一个字符变量
    string s="Hello World!"; //定义一个字符串变量
    char[] cArray={'H', 'e', 'l', 'l', 'o', ',', ' ', 'w', 'o', 'r', 'l', 'd' }; //定义一个字符数组
    Page p=new Page(); //定义一个 Page 对象
    Response.Write("输出单个字符");
    Response.Write(c);
    Response.Write("<br>");
    Response.Write("输出一个字符串"+s+"<br>");
    Response.Write("输出字符数组");
    Response.Write(cArray, 0, cArray.Length);
    Response.Write("<br>");
    Response.Write("输出一个对象");
    Response.Write(p);
    Response.Write("<br>");
    Response.Write("输出一个文件");
    Response.WriteFile(Server.MapPath(@"TextFile.txt"));
}
```

注意：输出一个文件时，该文件必须是已经存在的，如果不存在将产生“未能找到文件”异常。