

ASP.NET

项目案例教程



主编：刘贵君

副主编：戴涛 徐泓 付兴宏

 武汉理工大学出版社
WUTP Wuhan University of Technology Press

ASP.NET 项目案例教程

主 编 刘贵君
副主编 戴 涛 徐 泓 付兴宏

武汉理工大学出版社
· 武 汉 ·

ISBN 7-302-10001-1
定价：35.00元

· 内容提要 ·

【内 容 提 要】

本书基于 Visual studio 2010 集成开发工具,系统全面地介绍了最新 ASP.NET4.0 的技术设计、开发和部署 Web 网站的相关知识。

全书共 6 章,内容包括 ASP.NET Web 应用程序设计概述、ASP.NET 内部对象、ADO.NET、Web 服务器控件、应用程序配置、网上书店网站综合实例等内容。

本书运用软件工程的基本理论,结合面向对象的思想,采用项目教学,将 ASP.NET 相关知识融入项目案例中。

本书可作为普通高校应用型本科和高职高专类院校的程序设计教程,也可作为编程开发人员培训教材或供相关人员自学使用。

图书在版编目(CIP)数据

ASP.NET 项目案例教程 / 刘贵君主编. — 武汉:武汉理工大学出版社, 2017.6

ISBN 978-7-5629-5558-0

I. ①A… II. ①刘… III. ①网页制作工具-程序设计-教材 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2017)第 127145 号

项目负责人:陈 硕

责任编辑:陈 硕

责任校对:张莉娟

封面设计:付 群

出版发行:武汉理工大学出版社

社 址:武汉市洪山区珞狮路 122 号

邮 编:430070

网 址:<http://www.wutp.com.cn>

经 销:各地新华书店

印 刷:湖北丰盈印务有限公司

开 本:787×1092 1/16

印 张:19.25

字 数:492 千字

版 次:2017 年 6 月第 1 版

印 次:2017 年 6 月第 1 次印刷

印 数:1—1000 册

定 价:40.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。

本社购书热线电话:027-87515778 87515848 87785758 87165708(传真)

· 版权所有 盗版必究 ·

前 言

为了适应高等院校应用型人才培养快速发展的趋势,以就业为导向,培养技能型、应用型的高级软件开发人员,着力提高学生的动手能力和创新能力,特编写了本书。

ASP.NET 是 Microsoft.NET 的一部分,是 Active Server Page(简称 ASP)的另一个版本;ASP.NET 提供了一个统一的 Web 开发模型,其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容,同时还提供了一种新的编程模型和结构,可生成伸缩性和稳定性更好的应用程序,并提供更好的安全保护。

本书共 6 章,由浅入深、系统全面地介绍了如何使用 ASP.NET 技术构建 Web 应用程序的相关知识,覆盖了 ASP.NET 常用知识点和大部分新技术。

第 1 章 ASP.NET Web 应用程序设计概述,主要介绍了 ASP.NET 应用程序的概念、Visual Studio 2010 开发环境,静态网页与动态网页的区别以及网页工作原理(戴涛编写,哈尔滨工程大学)。

第 2 章 ASP.NET 内部对象,主要介绍了 ASP.NET 提供的基本对象,主要有 Request 对象、Response 对象、Session 对象、Application 对象和 Server 对象(徐泓编写,黑龙江农垦总局总医院)。

第 3 章 ADO.NET,主要介绍 ADO.NET 是一组向 .NET Framework 程序员公开数据访问服务的类。ADO.NET 为创建分布式数据共享应用程序提供了一组丰富的组件。它提供了对关系数据、XML 和应用程序数据的访问,是 .NET Framework 中不可缺少的一部分。ADO.NET 支持多种开发需求,包括创建由应用程序、工具、语言或 Internet 浏览器使用的前端数据库客户端和中间层业务对象(付兴宏编写,辽阳职业技术学院)。

第 4 章 Web 服务器控件,主要介绍了 HTML 服务器控件、Web 服务器控件,并验证某个值是否在限定范围之内等(刘贵君编写,黑龙江工商学院)。

第 5 章 应用程序配置,主要介绍应用程序配置相关知识(刘贵君编写)。

第 6 章 网上书店网站综合实例,通过项目案例将所学的知识 and 课外的知识结合起来,利用完整的案例使学生掌握软件开发的流程以及相关知识的运用(刘贵君编写)。

本书运用软件工程的基本理论,结合面向对象的思想,采用项目教学法将 ASP.NET 相关知识融入到项目案例中。

全书由刘贵君统稿。本书在编写过程中得到了杜凯教授、周洪玉教授的支持和帮助,在此对他们表示感谢。

由于时间仓促,加之水平有限,书中难免有疏漏和错误,恳请读者批评指正。

编 者
2017 年 2 月

目 录

1 ASP.NET Web 应用程序设计概述	1
1.1 Web 技术概述	1
1.2 ASP.NET 与 Visual Studio 2010 开发平台	2
1.3 第一个 ASP.NET 应用程序	2
2 ASP.NET 内部对象	4
2.1 内部对象概述	4
2.2 Page 对象	4
2.3 Response 对象	5
2.4 Request 对象	8
2.5 Cookie 对象	12
2.6 Session 对象	14
2.7 Application 对象	15
2.8 Server 对象	16
3 ADO.NET	17
3.1 数据库服务器	17
3.2 SQL 语言基础	17
3.3 ADO.NET	19
3.4 数据绑定	29
4 Web 服务器控件	31
4.1 服务器控件概述	31
4.2 常用标准控件	31
4.3 文件上传控件	38
4.4 MultiView 与 View 控件	39
4.5 容器控件	39
4.6 验证控件	40
4.7 导航控件和用户控件	47
4.8 数据绑定控件与数据源控件	66
4.9 母版页、主题与皮肤	110
5 应用程序配置	126
5.1 概述	126
5.2 Web.config 文件结构	127
5.3 存储和访问数据库连接串	128

5.4 存储和访问自定义数据	130
6 网上书店网站综合实例	131
6.1 网上书店的数据库表及数据库关系图	131
6.2 系统架构	133
6.3 实现网上书店	199
参考文献	300

1	ASP.NET 网站开发案例教程	1
2	ASP.NET 网站开发案例教程	2
3	ASP.NET 网站开发案例教程	3
4	ASP.NET 网站开发案例教程	4
5	ASP.NET 网站开发案例教程	5
6	ASP.NET 网站开发案例教程	6
7	ASP.NET 网站开发案例教程	7
8	ASP.NET 网站开发案例教程	8
9	ASP.NET 网站开发案例教程	9
10	ASP.NET 网站开发案例教程	10
11	ASP.NET 网站开发案例教程	11
12	ASP.NET 网站开发案例教程	12
13	ASP.NET 网站开发案例教程	13
14	ASP.NET 网站开发案例教程	14
15	ASP.NET 网站开发案例教程	15
16	ASP.NET 网站开发案例教程	16
17	ASP.NET 网站开发案例教程	17
18	ASP.NET 网站开发案例教程	18
19	ASP.NET 网站开发案例教程	19
20	ASP.NET 网站开发案例教程	20
21	ASP.NET 网站开发案例教程	21
22	ASP.NET 网站开发案例教程	22
23	ASP.NET 网站开发案例教程	23
24	ASP.NET 网站开发案例教程	24
25	ASP.NET 网站开发案例教程	25
26	ASP.NET 网站开发案例教程	26
27	ASP.NET 网站开发案例教程	27
28	ASP.NET 网站开发案例教程	28
29	ASP.NET 网站开发案例教程	29
30	ASP.NET 网站开发案例教程	30
31	ASP.NET 网站开发案例教程	31
32	ASP.NET 网站开发案例教程	32
33	ASP.NET 网站开发案例教程	33
34	ASP.NET 网站开发案例教程	34
35	ASP.NET 网站开发案例教程	35
36	ASP.NET 网站开发案例教程	36
37	ASP.NET 网站开发案例教程	37
38	ASP.NET 网站开发案例教程	38
39	ASP.NET 网站开发案例教程	39
40	ASP.NET 网站开发案例教程	40
41	ASP.NET 网站开发案例教程	41
42	ASP.NET 网站开发案例教程	42
43	ASP.NET 网站开发案例教程	43
44	ASP.NET 网站开发案例教程	44
45	ASP.NET 网站开发案例教程	45
46	ASP.NET 网站开发案例教程	46
47	ASP.NET 网站开发案例教程	47
48	ASP.NET 网站开发案例教程	48
49	ASP.NET 网站开发案例教程	49
50	ASP.NET 网站开发案例教程	50
51	ASP.NET 网站开发案例教程	51
52	ASP.NET 网站开发案例教程	52
53	ASP.NET 网站开发案例教程	53
54	ASP.NET 网站开发案例教程	54
55	ASP.NET 网站开发案例教程	55
56	ASP.NET 网站开发案例教程	56
57	ASP.NET 网站开发案例教程	57
58	ASP.NET 网站开发案例教程	58
59	ASP.NET 网站开发案例教程	59
60	ASP.NET 网站开发案例教程	60
61	ASP.NET 网站开发案例教程	61
62	ASP.NET 网站开发案例教程	62
63	ASP.NET 网站开发案例教程	63
64	ASP.NET 网站开发案例教程	64
65	ASP.NET 网站开发案例教程	65
66	ASP.NET 网站开发案例教程	66
67	ASP.NET 网站开发案例教程	67
68	ASP.NET 网站开发案例教程	68
69	ASP.NET 网站开发案例教程	69
70	ASP.NET 网站开发案例教程	70
71	ASP.NET 网站开发案例教程	71
72	ASP.NET 网站开发案例教程	72
73	ASP.NET 网站开发案例教程	73
74	ASP.NET 网站开发案例教程	74
75	ASP.NET 网站开发案例教程	75
76	ASP.NET 网站开发案例教程	76
77	ASP.NET 网站开发案例教程	77
78	ASP.NET 网站开发案例教程	78
79	ASP.NET 网站开发案例教程	79
80	ASP.NET 网站开发案例教程	80
81	ASP.NET 网站开发案例教程	81
82	ASP.NET 网站开发案例教程	82
83	ASP.NET 网站开发案例教程	83
84	ASP.NET 网站开发案例教程	84
85	ASP.NET 网站开发案例教程	85
86	ASP.NET 网站开发案例教程	86
87	ASP.NET 网站开发案例教程	87
88	ASP.NET 网站开发案例教程	88
89	ASP.NET 网站开发案例教程	89
90	ASP.NET 网站开发案例教程	90
91	ASP.NET 网站开发案例教程	91
92	ASP.NET 网站开发案例教程	92
93	ASP.NET 网站开发案例教程	93
94	ASP.NET 网站开发案例教程	94
95	ASP.NET 网站开发案例教程	95
96	ASP.NET 网站开发案例教程	96
97	ASP.NET 网站开发案例教程	97
98	ASP.NET 网站开发案例教程	98
99	ASP.NET 网站开发案例教程	99
100	ASP.NET 网站开发案例教程	100

1 ASP.NET Web 应用程序设计概述



- 静态网页与动态网页
- 网页工作原理

1.1 Web 技术概述

1.1.1 静态网页与动态网页

目前的 Web 页面根据其生成方式,可以分为静态网页和动态网页。

静态网页是指用 HTML 承载需要显示信息的 Web 页面。网页文件中不包含服务器端程序代码,文件的扩展名为 .htm 或 .html。静态网页发布到网站上,内容不会变化。如果需要修改内容,则需要修改静态网页的源代码并重新发布到网站上,内容才会变化。单纯使用静态网页技术建设网站,在早期较为流行,虽然网页中包含文字和图片等,但只要不改变设计,网页的显示内容就不会变化。

动态网页不仅包含 HTML 标记,而且包含服务器端程序代码。动态网页的后缀一般根据不同的程序设计语言而不同,如 ASP.NET 文件的后缀为 .aspx。动态网页会根据程序代码逻辑显示不同的内容,如张三和李四的 163 邮箱里显示的内容不同。

1.1.2 网页工作原理

1. 静态网页工作原理

当在客户端浏览器里输入一个静态网页网址并回车后,就向万维网服务器提出了一个浏览网页的请求。服务器端接到请求后,查找到用户请求的页面,然后发给客户端。

2. 动态网页工作原理

动态网页的工作原理和静态网页有很大的不同。当在浏览器地址栏中输入一个动态网页网址并回车后,就向服务器提出了一个浏览网页的请求。服务器端接收到请求后,首先查找到此动态网页,然后运行网页文件中的程序代码,将含有程序代码的动态网页转化为标准的静态网页后,再将静态网页发送客户端。

1.2 ASP.NET 与 Visual Studio 2010 开发平台

ASP.NET 的前身是 ASP(Active Server Pages)。ASP 的第一个正式版本 ASP1.0 是作为 IIS(Internet 信息服务)的附属产品免费赠送的,目前最高版本为 3.0。ASP 文件就是在普通的 HTML 文件中嵌入 VBScript 或 JavaScript 脚本语言。当客户请求一个 ASP 文件时,服务器端就会运行 ASP 文件中的脚本代码,并翻译为标准的 HTML 文件,然后发送给客户端。ASP 提供了几个非常有用的内部对象和内部组件,利用它们可以轻松地实现表单上传、存储数据库数据等功能。

ASP.NET 不是 ASP 的简单升级,而是 Microsoft 提出的 .NET 框架的一部分,它是一种以 .NET 框架为基础的开发 Web 应用的全新模式。

2002 年,Microsoft 公司正式在中国推出了 Visual Studio .NET 中文版,其中使用的 .NET 框架为 1.0 版本,ASP.NET1.0 也诞生了。2003 年,Microsoft 公司发布了 Visual Studio.NET 2003,其 .NET 框架为 1.1 版本。2005 年,Microsoft 公司发布了 Visual Studio 2005,其 .NET 框架为 2.0 版本。2007 年年底,Microsoft 公司发布了 Visual Studio 2008,其 .NET 框架为 3.0 版本或是 3.5 版本。ASP.NET3.5 版本和 2.0 版本相比,增加了一些新功能和一些新的控件与技术,如 Ajax、Linq 技术及 ListView 控件、DataPager 控件等。基于 ASP.NET3.5 的 Web 应用程序一般是通过 Visual Studio 2008(简称 VS2008)进行开发的。VS2008 是一个功能强大的集成开发平台,几乎可以完成所有的软件开发任务,包括 Windows 应用程序开发、Web 应用程序开发、设备应用程序开发以及与其相关的数据库应用处理等,同时还提供了各种应用程序的发布与安装支持。本书将采用高更版本 VS2010 进行开发。

1.3 第一个 ASP.NET 应用程序

【例 1-1】 现在示范制作一个简单 Web 应用程序的完整过程。该程序的功能是显示网站的欢迎信息。

(1) 打开 VS2010 开发工具。

(2) 选择【文件】菜单中的【新建】子菜单中的【网站】菜单项,在弹出的【新建网站】对话框中,选择【ASP.NET 空网站】模板,设置网站存储文件路径,单击【确定】按钮,如图 1-1 所示。

(3) 创建新网站之后,打开的界面如图 1-2 所示。VS2010 开发工具的界面按从上到下、从左到右的顺序包括标题栏、菜单栏、工具栏、工具箱、页



图 1-1 新建 ASP.NET 网站

面设计窗口、解决方案资源管理器、错误列表窗口、属性窗口。

```
1 <% Page Language="CS" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7 <title></title>
8 </head>
9 <body>
10 <form id="form1" runat="server">
11 <div>
12
13 </div>
14 </form>
15 </body>
16 </html>
17
```

图 1-2 新建网站的默认窗口

主要窗口说明如下：

- 工具箱放置着开发 ASP.NET 页面所需的各种页面控件，拖拉工具箱中的控件到页面中即可生成该控件。
- 错误列表窗口，主要用于显示设计页面或是代码中出现的语法错误，可以方便程序员及时更正程序中的各种语法错误。
- 解决方案资源管理器，主要显示网站中的各个文件，通过该窗口可以快速浏览目录和文件、创建网站需要的各种文件。
- 属性窗口，当选择页面中任何一个控件时，属性窗口将自动显示该控件的相关属性，并且可以在这个窗口中设置控件的属性。
- 页面设计窗口，主要用于对每个页面进行相关的设计，单击其下的【设计】、【拆分】、【源】按钮，可以在页面的设计视图与源代码视图之间进行切换，“拆分”指页面的上半部分是源代码视图，下半部分是设计视图。

(4) 单击页面设计窗口下部的【设计】按钮切换到设计视图，单击鼠标左键，选择标签栏上的“<div>”标签。

(5) 在属性窗口中，选择“Style”属性，单击其右边的按钮，在弹出的对话框中设置“font-size”属性为 20pt。

(6) 在设计视图窗口的蓝色矩形 div 标记中输入“欢迎您来到校园音乐吧”，如图 1-3 所示。

欢迎您来到校园音乐吧

图 1-3 新建网站的默认窗口

(7) 单击工具栏上的【启动调试】按钮或按“F5”键运行此 Web 程序。

2 ASP.NET 内部对象



本章要点

- 掌握 Page 对象的 IsPostBack 属性及 Load 事件的用法
- 掌握 Response 对象的 Write 方法和 Redirect 方法的用法
- 掌握 Request 对象读取数据的方法
- 掌握 Cookie 对象的设置与读取
- 掌握 Session 对象的用法
- 掌握 Application 对象的用法
- 掌握 Server 对象的用法

2.1 内部对象概述

ASP.NET 提供的基本对象主要有 Request 对象、Response 对象、Session 对象、Application 对象和 Server 对象。其中 Request 对象产生于 HttpRequest 类,该类使得 ASP.NET 能够读取客户端在 Web 请求期间发送的 HTTP 值;Response 对象产生于 HttpResponse 类,该类封装了来自 ASP.NET 操作的 HTTP 响应信息;Session 对象产生于 HttpSessionState 类,该类提供对会话状态值及会话级别设置和生存期管理方法的访问;Application 对象产生于 HttpApplication 类,该类定义了 ASP.NET 应用程序中的所有应用程序对象通用的方法、属性和事件,此类是用户在 "global.asax" 文件中所定义的应用程序的基类;Server 对象产生于 HttpServerUtility 类,该类提供了用于处理 Web 请求的方法,HttpServerUtility 类的方法和属性通过 ASP.NET 提供的内部 Server 对象公开。

2.2 Page 对象

Page 对象指的是页面对象,是由 System.Web.UI.Page 类实现。Page 对象的常用属性、事件和方法如表 2-1 所示。

表 2-1 Page 对象的属性、事件和方法

名 称	说 明
IsPostBack	属性,表示页面上的表单是否被提交过,如提交为 true,否则为 false
IsValid	属性,表示页面上所有的控件是否通过验证,如通过为 true,否则为 false
Load	属性,加载页面时发生
Unload	属性,卸载页面时发生

Page 对象的常用组合代码如下:

```
Protected void Page_Load(Object sender,EventArgs e)
{
    If(!IsPostBack)
    {
        首次打开页面进入执行的代码
    }
}
```

! IsPostBack 表示表单没有被提交过,即页面首次被加载,因此可以把只在页面打开时执行的代码放在 if 语句中。

2.3 Response 对象

Response 对象是 HttpResponse 类的一个实例。该类主要是封装来自 ASP.NET 操作的 HTTP 响应信息。

2.3.1 Response 对象的属性(表 2-2)

表 2-2 Response 对象的属性

属 性	说 明	属 性 值
BufferOutput	获取或设置一个值,该值指示是否缓冲输出,并在完成处理整个页面之后将其发送	如果缓冲到客户端的输出,则为 true; 否则为 false。默认为 true
Cache	获取 Web 页的缓存策略(过期时间、保密性、变化子句)	包含有关当前响应的缓存策略信息的 HttpCachePolicy 对象
Charset	获取或设置输出流的 HTTP 字符集	输出流的 HTTP 字符集
IsClientConnected	获取一个值,通过该值指示客户端是否仍连接在服务器上	如果客户端当前仍在连接,则为 true; 否则为 false

【例 2-1】 使用缓冲区

由于 Response 对象的 BufferOutput 属性默认为 true,所以要输出到客户端的数据都暂

时存储在缓冲区内,等到所有的事件程序以及所有的页面对象全部解译完毕后,才将所有在缓冲区中的数据送到客户端的浏览器。下面的例子将演示缓冲区是如何工作的。

```
<%Response.Write("缓存已清除"+"<Br>");%>
<Script Language="C#" Runat="Server">
void Page_Load(Object sender, EventArgs e)
{
    Response.Write("缓存清除前"+"<Br>");
    Response.Clear();
}
</Script>
```

上述程序代码实例首先在“Page_Load”事件中送出“缓存清除前”这一行,此时的数据存储在缓冲区中。接着使用 Response 对象的 Clear 方法将缓冲区的数据清除,所以刚刚送出的字符串已经被清除。然后 IIS 开始读取 HTML 组件的部分,并将结果送至客户端的浏览器。由执行结果只出现“缓存已清除”可知,使用 Clear 方法之前的数据并没有出现在浏览器上,所以程序开始时是存储在缓冲区内的。如果在相同的程序中加入“Response. BufferOutput = false”,有:

```
<%Response.Write("清除之后的数据<Br>");%>
<Script Language="C#" Runat="Server">
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();
}
</Script>
```

可以发现,执行的结果并没有因为使用 Clear 方法而将缓冲区的数据清除,这表明数据是直接输出而没有存放在缓冲区内。

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。表 2-3 列举了几个常用的方法。

表 2-3 Response 对象的方法

方 法	说 明
Write	将指定的字符串或表达式的结果写到当前的 HTTP 输出
End	停止页面的执行并得到相应结果
Clear	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
Flush	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
Redirect	使浏览器立即重定向到程序指定的 URL

2.3.2 利用 Write 方法输出信息

Response.Write(字符串):将字符串以 HTML 的格式送到客户端显示。

【例 2-2】 使用 Response.write 方法向客户端发送信息

```
for(int i=1;i<=500;i++)
{
    Response.Write("i="+i+"<BR>");
}
```

本例使用“write”方法,向屏幕输出 500 个值。

【例 2-3】 使用 Response.End 方法调试程序

End 方法可以停止当前页面的执行,基于这个原因,可以结合 Response.write 方法输出当前页面上的某个变量、数组值。

```
<form id="Form1" method=" post" runat="server">
    输入一个数值:
    <asp:TextBox id="txtVar" runat="server"></asp:TextBox>
    <asp:Button id="btnSubmit" runat="server" Text="计算该值的平方值"
onclick="btnSubmit_Click"></asp:Button>
</form>
<Script Language="C#" Runat="Server">
    void btnSubmit_Click(Object sender, EventArgs e)
    {
        int N=int.Parse(Request.Form["txtVar"].ToString());
        Response.Write("N="+N+"<br>");
        Response.Write("该值的平方值是:"+N*N);
    }
</Script>
```

输入一个值“6”,然后单击【计算该值的平方值】按钮,屏幕将显示如下结果:N=6,该值的平方值是 36。

在代码中加上“Response.End()”,代码如下:

```
<Script Language="C#" Runat="Server">
    void btnSubmit_Click(Object sender, EventArgs e)
    {
        int N=int.Parse(Request.Form["txtVar"].ToString());
        Response.Write("N="+N+"<br>");
        Response.End();
        Response.Write("该值的平方值是:"+N*N);
    }
</Script>
```

这时再运行代码,将只会显示:N=6。实验证明,“Response.End()”方法停止了当前页面的执行。这仅仅是一个小例子,读者可以依次类推,在程序中使用 End 方法进行调试。不过千万要记住调试完代码后,不要忘记把调试用的“Response.End()”删掉。

2.3.3 利用 Redirect 方法实现页重定向

在网页编程中,经常会遇到在程序执行到某个位置进行页面调转的情况。Response.Redirect 方法可以满足这种需求。

Response.Redirect(URL):将浏览器重定向到另一个网页

例如代码:Response.Redirect("http://www.163.com");执行该代码,页面将跳转到网易 163 的主页。

2.4 Request 对象

Request 对象是 HttpRequest 类的一个实例。它能够读取客户端在 Web 请求期间发送的 HTTP 值。

Request["变量名"]:取得页面之间传递变量的值。

Request 对象的属性如表 2-4 所示。

表 2-4 Request 对象的属性

属 性	说 明	属 性 值
QueryString	获取 HTTP 查询字符串变量集合	NamueCollection 对象
Path	获取当前请求的虚拟路径	当前请求的虚拟路径
UserHostAddress	获取远程客户端的 IP 主机地址	远程客户端的 IP 地址
Browser	获取有关正在请求的客户端的浏览器功能的信息	HttpBrowserCapabilities 对象

2.4.1 获取 QueryString 数据集合中的数据

【例 2-4】 获取 QueryString 值

程序中经常可以使用 QueryString 来获得从上一个页面传递来的字符串参数。例如,在页面 1 中创建一个连接,指向页面 2,并用 QueryString 来查询两个变量:查看

在页面 2 中接收到从页面 1 传过来的两个变量:

```
<Script Language="C#" Runat="Server">
    void Page_Load(object sender, System.EventArgs e)
    {
        Response.Write("变量 ID 的值:"+Request.QueryString["ID"] + "<br>");
        Response.Write("变量 Name 的值:"+
            Ruest.QueryString["Name"]);
    }
```

```
</Script>
```

运行上面代码结果如下：

变量 ID 的值:6

变量 Name 的值:Wang

上面的例子可以成功地得到 QueryString 的值。用类似方法,可以获取 Form, Cookies, ServerVariables 的值。调用方法都是: Request. Collection[" Variable"], Collection 包括 QueryString, Form, Cookies, ServerVariables 四种集合, Variable 为要查询的关键字。不过,这里的 Collection 是可以省略的,也就是说, Request[" Variable"]与 Request. Collection["Variable"]这两种写法都是允许的。如果省略了 Collection,那么 Request 对象会依照 QueryString, Form, Cookies, ServerVariables 的顺序查找,直至发现 Variable 所指的关键字并返回其值,如果没有发现其值,则返回空值(null)。不过,为了优化程序的执行效率,建议最好还是使用 Collection,因为过多地搜索就会降低程序的执行效率。

2.4.2 获取 Form 表单中的数据

```
<%@Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script runat="server">
        protected void Page_Load(object sender, EventArgs e)
        {
            string name=Request.Form["name"];
            string website=Request.Form["website"];
            Response.Write("你的公司:"+name+"<br/>"+"你的网址:"+website+"<br/>");
            Response.Write("你使用的是 "+Request.RequestType
+" 方式传送数据。");
            Response.Write("<br/>");
            Response.Write("当前页面调用状态:"+IsPostBack+"
+IsPostBack+"<br/>");
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <p>
                默认原始请求 (HTTP GET) 方式发送数据:<br />
            </p>
            <p>
```

```

        你的公司
        <asp:TextBox ID="name" runat="server"
            Text="Google">
        </asp:TextBox><br />
        你的网站
        <asp:TextBox ID="website" runat="server"
            Text="www.google.com.hk">
        </asp:TextBox><br />
    </p>
    <input type="submit" value="POST 方式发送" />
    <br />
</div>
</form>
</body>
</html>
```

2.4.3 获取 ServerVariables 中的数据

利用 Request 对象的 ServerVariables 数据集合可以方便地取得服务器端和客户端的环境变量信息,常见的环境变量如表 2-5 所示。

表 2-5 常见的环境变量

环境变量名称	说 明
App_Physical_Path	应用程序实际路径
Local_Addr	服务器端的 IP 地址
Remote_Addr	客户端的 IP 地址
Server_Name	服务器端计算机名
Remote_Host	客户端计算机名
Server_Port	服务器端口号
Query_String	http 请求中“?”号后的内容
Content_Length	客户端发送的内容的长度,以字节为单位
Content_Type	客户端请求的 MIME 内容类型

获取环境变量值的语法为:Request.ServerVariables["环境变量"]

下面代码用来显示服务器名字和客户端 IP 地址。

```
protected void Page_Load(object sender, EventArgs e)
{
    String strServerName =
        Request.ServerVariables["Server_Name"];
    Response.Write(strServerName+"<br/>");
}
```

```
String strIP=Request.ServerVariables["Remote_Addr"];
Response.Write(strIP);
}
```

2.4.4 使用 Response 对象和 Request 对象

【例 2-5】 使用 Response 对象和 Request 对象在不同页面之间的传递变量。

具体步骤：

(1) Login.aspx 文件：

登录页面如图 2-1 所示。

注意：TextMode="Password" 表示为密码

(2) Login.aspx.cs 文件

登录按钮的事件处理代码：

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

```
{
```

```
    string name=txtUserName.Text;
```

```
    string pwd=txtPassword.Text;
```

```
    bool flag=false;
```

```
    flag=true;
```

```
    if (flag)
```

```
    {
```

```
        Response.Redirect ("LoginHandle.aspx?username="+
                                name+"&password="+pwd);
```

```
    }
```

```
    else
```

```
    {
```

```
        Response.Write ("登录失败!");
```

```
    }
```

```
}
```

(3) 建立一个处理文件 LoginHandle.aspx

其中 LoginHandle.aspx.cs 的代码如下：

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

```
{
```

```
    Response.Write ("用户名："+Request ["username"]);
```

```
    Response.Write ("<br>");
```

```
    Response.Write ("密码："+Request ["password"]);
```

```
}
```

说明：

(1) 超链接传递参数：文件名？参数名 1=参数值 & 参数名 2=参数值 2&...；如果需传递多个变量，则多个变量之间需要用 & 连接。

(2) Request ["username"] 用来接传过来的参数值。

图 2-1 登录页面