

相关主题

RECOMMEND ARTICLE

- ▶ 在ASP.NET 2.0中建立站点导航层次
- ▶ JSP和JSF双剑合并 打造完美Web应用
- ▶ 用ASP.NET2.0在数据库中存储二进制文件
- ▶ ASP.NET中上传文件到数据库
- ▶ ASP.NET定制简单的错误处理页面
- ▶ 基于JSF技术的WEB应用开发研究
- ▶ ASP.NET实现投票结果的图片进度条显示
- ▶ 圣殿骑士PHP 2007年Web开发技术预言

[MORE](#)

推荐文章

RECOMMEND ARTICLE

- ▶ 数据广播方案的优化
- ▶ 网络游戏的位置同步
- ▶ 游戏音乐制作案例之《战火 红色警戒》音效制作揭秘
- ▶ 英雄连Online 原画
- ▶ 游戏音乐制作案例之《乱武天下》
- ▶ 游戏音乐制作案例之《诛仙》
- ▶ 《鹿鼎记》最新原画
- ▶ MIDP2.1规范的新特性

[MORE](#)

热门文章

HOT ARTICLE

- ▶ [电子书下载]游戏设计 — 原理与实践
- ▶ [电子书下载]网络游戏开发
- ▶ 游戏设计全过程
- ▶ [电子书下载]游戏设计技术
- ▶ [电子书下载]游戏设计理论
- ▶ CS游戏人物模型制作教程
- ▶ CG人物插画基本流程
- ▶ [转贴]MAX高级人头教程

[MORE](#)

您的位置: WEB技术



文章标题	ASP.NET四种页面导航方式之比较与选择		
来源:	[ogdev]	浏览:	[506]

在ASP.NET应用中, Web表单之间的导航有多种方式: 用超级链接, 用Response.Redirect, 用Server.Transfer, 或者用Server.Execute。本文将分析这四种导航方式的异同及其优缺点, 帮助你选择最佳的导航方式。

一、超级链接

从一个表单进入另一个表单最简单的方式是使用HTML超级链接控件。在Web表单中, 使用超级链接的HTML代码类如:

进入表单2

当用户点击该超级链接, WebForm2.aspx执行并将结果发送到浏览器。超级链接导航方式几乎可用于任何地方, 包括HTML页面和普通的ASP页面。ASP.NET还提供了另一种可替换使用的方法, 即HyperLink服务器控件:

```
NavigateUrl="WebForm2.aspx">进入表单2
```

上述HTML代码的运行结果和第一个例子相同, 因为ASP.NET把HyperLink Web服务器控件视为一个HTML超级链接控件。但两者有一点重要的区别, HyperLink Web服务器控件可以在服务器端编程。具体地说, 可以在程序代码中改变它的NavigateUrl属性, 从而允许构造出具体目标可根据应用的当前状态动态变化的超级链接, 例如:

```
Private Sub Button1_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles Button1.Click
    HyperLink1.NavigateUrl = "WebForm3.aspx"
End Sub
```

这段代码执行后, 如果用户点击链接, 他看到的将是WebForm3.aspx, 而不是WebForm2.aspx。

二、用程序控制重定向

虽然超级链接能够从一个页面导航到另一个页面, 但这种导航方式是完全由用户控制的。有些时候, 我们可能要用代码来控制整个导航过程, 包括何时转到另一个页面。在这些场合, ASP.NET有三种不同的方式可以达到相似的目的: 调用Response对象的Redirect方法, 调用Server对象的Transfer或Execute方法。这三种导航方式的行为基本相似, 但也有区别。

2.1 Response.Redirect

Response.Redirect方法导致浏览器链接到一个指定的URL。当Response.Redirect()方法被调用时, 它会创建一个应答, 应答头中指出了状态代码302(表示目标已经改变)以及新的目标URL。浏览器从服务器收到该应答, 利用应答头中的信息发出一个对新URL的请求。

这就是说, 使用Response.Redirect方法时重定向操作发生在客户端, 总共涉及到两次与服务器的通信(两个来回): 第一次是对原始页面的请求, 得到一个302应答, 第二次是请求302应答中声明的新页面, 得到重定向之后的页面。

2.2 Server.Transfer

Server.Transfer方法把执行流程从当前的ASPX文件转到同一服务器上的另一个ASPX页面。调用Server.Transfer时, 当前的ASPX页面终止执行, 执行流程转入另一个ASPX页面, 但新的ASPX页面仍使用前一ASPX页面创建的应答流。

如果用Server.Transfer方法实现页面之间的导航, 浏览器中的URL不会改变, 因为重定向完全在服务器端进行, 浏览器根本不知道服务器已经执行了一次页面变换。

默认情况下, Server.Transfer方法不会把表单数据或查询字符串从一个页面传递到另一个页面, 但只要把该方法的第二个参数设置成True, 就可以保留第一个页面的表单数据和查询字符串。

同时, 使用Server.Transfer时应注意一点: 目标页面将使用原始页面创建的应答流, 这导致ASP.NET的机器验证检查(Machine Authentication Check, MAC)认为新页面的ViewState已被篡改。因此, 如果要保留原始页面的表单数据和查询字符串集合, 必须把目标页面Page指令的EnableViewStateMac属性设置成False。

2.3 Server.Execute

Server.Execute方法允许当前的ASPX页面执行一个同一Web服务器上的指定ASPX页面，当指定的ASPX页面执行完毕，控制流程重新返回原页面发出Server.Execute调用的位置。

这种页面导航方式类似于针对ASPX页面的一次函数调用，被调用的页面能够访问发出调用页面的表单数据和查询字符串集合，所以要把被调用页面Page指令的EnableViewState属性设置成False。

默认情况下，被调用页面的输出追加到当前应答流。但是，Server.Execute方法有一个重载的方法，允许通过一个TextWriter对象（或者它的子对象，例如StringWriter对象）获取被调用页面的输出，而不是直接追加到输出流，这样，在原始页面中可以方便地调整被调用页面输出结果的位置。

为说明其工作过程，下面我们创建一个Web表单，放入一个按钮控件（Button1）和一个文本控件（Literal1），在设计界面中转入代码视图，加入一个System.IO名称空间的Imports语句，然后加入用户点击按钮时执行的代码：

```
Private Sub Button1_Click( _  
ByVal sender As System.Object, _  
ByVal e As System.EventArgs) _  
Handles Button1.Click  
Dim sw As StringWriter = New StringWriter()  
Server.Execute("WebForm2.aspx", sw)  
Literal1.Text = sw.ToString()  
End Sub
```

然后为同一个Web应用创建第二个页面WebForm2.aspx。转入该页面的HTML视图，修改其Page指令禁止ViewState检查：

再转到设计视图，为第二个页面增加一些控件。接下来，把第一个页面设置成默认页面，启动应用。点击按钮，WebForm2的控件将显示在WebForm1中放置Literal按钮的地方，注意页面标题和URL仍旧显示原始页面WebForm1。



用Server.Transfer或Server.Execute方法实现导航时，还要注意一点：最后得到的页面可能不是合法的HTML页面，因为最终返回给客户端的页面可能包含多个和等标记。IE浏览器看来能够容忍并正确处理这类情形，但如果用户要用到其他的浏览器，最好仔细测试一下。

三、比较与选择

既然从一个页面导航到另一个页面的办法有这么多，应该如何选择最佳的导航方式呢？下面是一些需要考虑的因素：

- 如果要让用户来决定何时转换页面以及转到哪一个页面，超级链接最适合。
- 如果要用程序来控制转换的目标，但转换的时机由用户决定，使用Web服务器的HyperLink控件，动态设置其NavigateUrl属性。
- 如果要把用户连接到另一台服务器上的资源，使用Response.Redirect。
- 用Response.Redirect把用户连接到非ASPX的资源，例如HTML页面。
- 如果要将查询字符串作为URL的一部分保留，使用Response.Redirect。
- 如果要将执行流程转入同一Web服务器的另一个ASPX页面，应当使用Server.Transfer而不是Response.Redirect，因为Server.Transfer能够避免不必要的网络通信，从而获得更好的性能和浏览效果。
- 如果要捕获一个ASPX页面的输出结果，然后将结果插入另一个ASPX页面的特定位置，则使用Server.Execute。
- 如果要确保HTML输出合法，请使用Response.Redirect，不要使用Server.Transfer或Server.Execute方法。

本栏目登载此文出于传递信息之目的，如有任何的问题请及时和我们联系！

无任何评论！	
<p>请您注意：</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> 尊重网上道德，遵守中华人民共和国各项有关法律法规<input checked="" type="checkbox"/> 尊重网上道德，遵守中华人民共和国的各项有关法律法规<input checked="" type="checkbox"/> 承担一切因您的行为而直接或间接导致的民事或刑事法律责任<input checked="" type="checkbox"/> 中国网游研发中心新闻留言板管理人员有权保留或删除其管辖留言中的任意内容<input checked="" type="checkbox"/> 您在中国网游研发中心留言板发表的作品，中国网游研发中心有权在网站内转载或引用<input checked="" type="checkbox"/> 参与本留言即表明您已经阅读并接受上述条款	<p>发表评论：</p> <p>昵 称：<input type="text"/> <input type="button" value="GO"/></p> <p>联系EMAIL：<input type="text"/></p> <p>    </p> <p><input type="text"/></p>