

# .NET 中页间传递服务器控件值的方法

## Means of Translate Values of Server Control in Pages in .NET

刘智勇 (江苏 南京农业大学 210095)

**摘要:** 创建 WEB 应用时,经常需要在不同的页面间传递服务器控件的值。在本文中,根据 .NET 面向对象的编程方法及其内在特性,讨论了使用内联代码和代码隐藏文件两种设计模式在页面间传递数据的方法。利于开发人员对 .NET 的理解和使用。

**关键词:** .NET 服务器控件值 数据传递 页间传递

在 WEB 应用程序开发中,页面之间数据传递必不可少,.NET 环境中,可以使用 Session 对象、Application 对象以及 URL 参数传递等方法实现,对于服务器控件值的传递,.NET 提供了一种内在的、简洁的、完全面向对象的解决机制。在本文中,以 C# 为例对使用内联代码和使用代码隐藏文件两种设计模式对这一问题进行讨论。

### 1 使用内联代码的设计模式

内联代码的设计模式是指服务器端代码嵌入在 WEB 窗体页面中,使用该方式将值传递到另一个 WEB 窗体页时,首先要在 @ Page 指令中为该 WEB 页指定类名。然后,为要传递的每个值创建一个具有 get 访问器的属性,返回传递的值(如文本框的值)。使用 Server 对象的 Transfer 方法将页面重定向到接收传递值的 WEB 窗体页。

接收 WEB 窗体页上,在页的顶部添加 @ Reference,并将 Page 属性设置为发送页以引用发送页中声明的类。.NET 处理程序从 HttpContext 对象的 Handler 属性接收到 Http 请求后,生成所传递信息的类的实例,通过该实例对象的属性访问所传递的值。示例代码及代码分析如下。

#### 1.1 发送值的 WEB 窗体页:sourcepage.aspx

```
<% @ Page Language = "C#"
ClassName = " SourcePageClass" % > // 在 Web
窗体页顶部的 @ Page 指令中设置类名为源 Web 窗
体页指定类名
```

```
<html > <head >
<script runat = " server" >
public string TransName {
get { return first.Text; }
} // 为需传递到另一个 Web 窗体页的每个值都定
义一个具有 get 访问器的属性,该属性只返回要传递
的值
void ButtonClicked ( object sender, EventArgs
e) {
Server.Transfer ( " aimpage.aspx" );
} // 使用 Transfer 方法将应用程序的控制传
输到指定 URL 的目标 WEB 窗体页
</script >
</head > <body > <form runat = " server" >
First Name: < asp:TextBox id = " first" runat = "
server" / >
<br >
< asp:Button OnClick = " ButtonClicked" Text
= " 传递参数" runat = server / > // 当单击按钮时,激
发 Server.Transfer 方法
</form > </body > </html >
```

#### 1.2 接收值的 WEB 窗体页:aimpage.aspx

```
<% @ Page Language = "C#" % >
<% @ Reference Page = " sourcepage.aspx" %
> // 在页的顶部添加 @ Reference 指令并将 Page 属
性设置为源 Web 窗体页,以引用源 WEB 窗体页中
的类
<html > <head >
```

```

<script runat = " server" >
    SourcePageClass fp;//定义一个用以存储发送信息的 WEB 窗体页中定义的类的实例的变量
    void Page_Load() {
        if (! IsPostBack) {
            fp = ( SourcePageClass ) Context. Handler;
        }
    }
    //创建自定义 Page_Load 事件处理程序,将当前 HTTP 请求的 IHttpHandler 实现的对象分配给声明的变量,并使用 IsPostBack 属性确定是否是本页回发到其本身 </script > </head > <body >
    <form runat = " server" >
        Hello <% = fp. TransName % > //变量 fp 包含封装源 Web 窗体页中信息的类的实例,可使用该类的属性(源 WEB 窗体页要传递的信息)进行所需的处理
    </form > </body > </html >

```

## 2 使用代码隐藏文件的设计模式

隐藏代码文件的设计模式是指将服务器端运行代码保存在单独的与 WEB 窗体代码关联的页中,在 WEB 窗体页 @ Page 指令中添加指向在代码隐藏文件中声明的类的属性。在代码隐藏文件中添加服务器端运行代码,除此以外,发送方式与接收方式与内联代码设计模式一致。示例代码及分析如下。

### 2.1 发送值的 WEB 窗体页:sourcepage.aspx

```

<% @ Page Language = " C#" AutoEventWireup = " true" CodeFile = " sourcepage.aspx.cs" Inherits = " bbs_sourcepage" % > //CodeFile 属性指定代码隐藏文件名,Inherits 属性声明在代码隐藏文件中的类名
<html > <head > </head > <body >
<form id = " Form2" runat = " server" >
    TransName: <asp:TextBox id = " first" runat = " server" / > <br / >
    <asp:Button id = " Button1" OnClick = " ButtonClicked" Text = " Go to second page" runat = " server" / > //当按钮单击时,调用代码隐藏文件中的 ButtonClicked 事件,激发 Server.Transfer 方法

```

```

</form > </body > </html >

```

### 2.2 与发送页关联的代码隐藏文件:sourcepage.aspx.cs

```

using System;
public partial class bbs_sourcepage: System.Web.Ul. Page {
    //创建 WEB 窗体页 Inherits 属性声明的类
    protected void Page_Load( object sender, EventArgs e) { }
    public string FirstName {
        get { return first. Text;
        } //定义具有 get 访问器的属性,
    }
    protected void ButtonClicked ( object sender, EventArgs e) { Server. Transfer( " aimpage.aspx" );
    } //激发 Server.Transfer 方法,传递到 aimpage.aspx
}

```

接收页接收方式与内联代码设计模式基本一致,示例代码省略。

## 3 在程序中生成服务器控件的值在不同 WEB 页中传递的方法

创建 WEB 应用时,很多时候需要根据不同的条件在程序中动态生成控件,动态生成的控件的值在不同的 WEB 页间的传递主要不同点在于控件生成与控件读取的方法。以隐藏代码文件的设计模式为例,主要示例代码如下:

### 3.1 在程序中动态生成控件

```

protected void Page_Load ( object sender, EventArgs e)
{
    TextBox tbox = new TextBox ( ); //定义包含 TextBox 的变量 tbox
    tbox. ID = " last" ; //指定该控件的 ID 为 last
    Panel1. Controls. Add ( tbox ); //将该控件添加到 Panel1 中
}

```

### 3.2 为要传递的值定义具有 get 访问器的属性

```
public string LastName {
```

(下转第 108 页)

(上接第 110 页)

```
get{ TextBox  
    aimtbox = (TextBox)Panel1. FindControl("last");  
    //根据生成的控件的 ID 访问空间  
    return aimtbox. Text; } //得到控件的值  
}
```

#### 4 结束语

服务器控件是 .NET 开发 WEB 应用程序的一大新的特点,根据 .NET 面向对象编程的方法及其内在特性在页面间传递服务器控件的值,较传统的利用 Session、Request、全局变量等方法在编程上稍显复杂,但是这

种方法充分体现了面向对象的编程思想,有助于开发人员更好的理解和使用 .NET 进行 WEB 开发。

#### 参考文献

- 1 美 Stephen C. Perry 肖斌译,《C#和 .NET 核心技术 [M]》,北京:机械工业出版社,2006.
- 2 Adding Controls to a Web Forms Page Programmatically. <http://msdn2.microsoft.com/en-us/library/default.aspx>.
- 3 周静,《Asp 和 Asp. Net 页面间的数据传递方式探索 [J]》,《微计算机信息》,2006,15:114 - 115.