

# ASP.NET WebForm

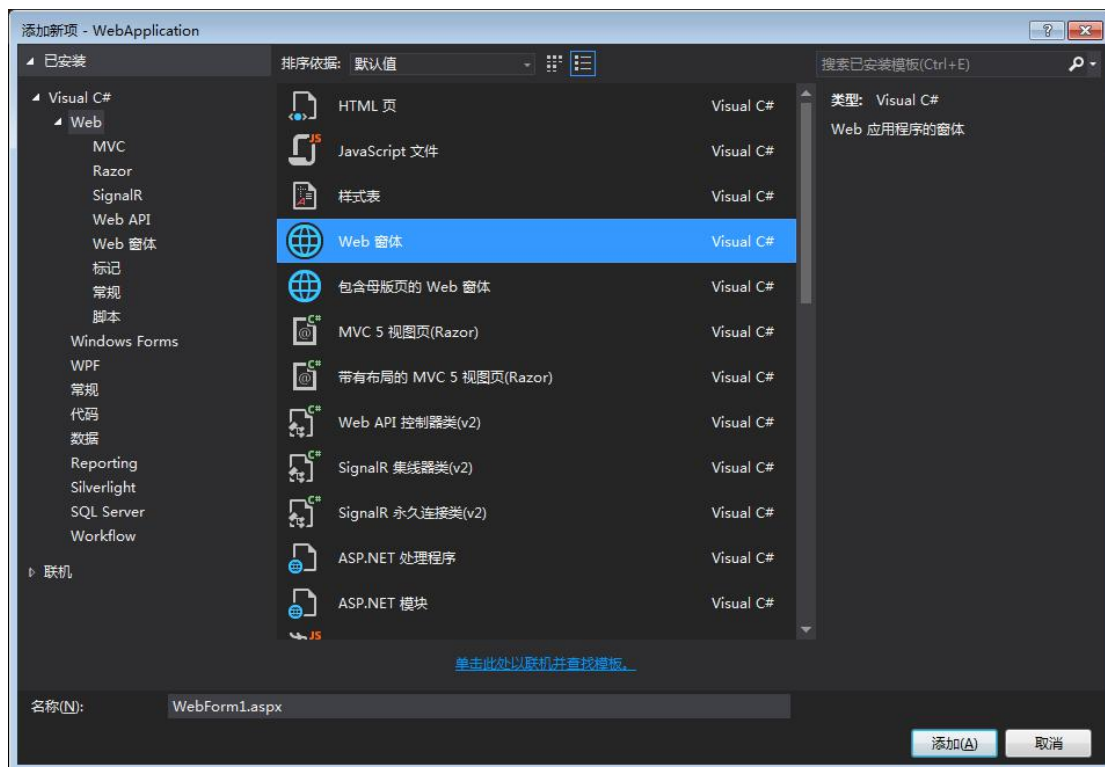
既然有了 Ashx，为什么还要有 Aspx？

如果每次输出网页都 一般处理程序(ashx) 的话太痛苦了，所以一般生成 html 的时候都直接创建 aspx ( Web 窗体，WebForm )

WebForm 分为两个文件 aspx 和 aspx.cs，aspx 是页面模板，是页面描述文件，就是 html+js+css 的内容 和 aspx.cs 结合的更好 不用像一般处理程序那样程序员自己去输出 HTML 字符串或读取填充模板，控件都是定义在 aspx 中，内联的 JavaScript、CSS 也是写在 aspx 中的，服务端的 C#代码是定义在 aspx.cs 中。aspx 控制页面长相，cs 控制程序逻辑，这种 “前 aspx 后 cs” 的方式就被称为 CodeBehind(代码后置)。

- cs 可以调用 aspx 中的 runat=server 的控件( 根本原因 )，aspx 中也可以访问 cs 中定义的非私有的成员(根本原因)，还可以编写复杂的 C#代码， for 等所有 C#代码都可以写在 aspx 中 ( 不推荐 )
- 前面     <%=UserName %>     <%=SayHello(); %>     <%if (UserName == "aaa")  
  
                 { UserName = "bbb"; } %>     后面
- 在当前位置输出表达式的值的时候使用<%=UserName %>，不要丢了=，相当于在当前位置调用 Response.Write(UserName)
- 使用的函数、代码相当于在这个位置调用函数、执行代码。注意 aspx 中调用 cs 的只能是非私有成员(非 private 修饰-因为继承关系)
- <%%>中的代码是运行在服务器端的，是 C#语法，在服务端运行完成才输出到浏览器，其他部分是运行在浏览器端的，是 html、JavaScript 语法

## 创建 Web 窗体



## 借助之前课程的三层访问所有数据

```
public partial class WebForm : System.Web.UI.Page
{
    //public string strHtml { set; get; }
    public List<UserInfo> UserInfoList { set; get; }
    protected void Page_Load(object sender, EventArgs e)
    {
        UserInfoService userInfoService = new UserInfoService();
        List<UserInfo> list = userInfoService.GetEntityList();
        UserInfoList = list;
        //StringBuilder sb = new StringBuilder();
        //foreach (UserInfo userInfo in list)
        //{
        //
        //
        //
        sb.AppendFormat("<tr><td>{0}</td><td>{1}</td><td>{2}</td><td>{3}</td><td>{4}</td><td><a href='UserInf.ashx?编号={5}'>详情</a><td><a href='delete.ashx?编号={5}' class='del'>删除</a><td><a href='update.ashx?编号={5}'>修改</a></td></tr>", userInfo.UserID, userInfo.UserName, userInfo.UserPass, userInfo.RegTime, userInfo.Email, userInfo.UserID);
        //}
        //strHtml = sb.ToString();
    }
}
```

```
}
```

```
<form id="form1" runat="server">
    <div>
        <a href="AddUserInfo.aspx">添加用户</a>
        <table>
            <tr>
                <th>编号</th>
                <th>用户名</th>
                <th>密码</th>
                <th>时间</th>
                <th>邮箱</th>
                <th>详情</th>
                <th>删除</th>
                <th>修改</th>
            </tr>
            <%=strHtml %>
            <%foreach (UserInfo item in UserInfoList){%>
                <tr>
                    <td><%=item.UserID %></td>
                    <td><%=item.UserName %></td>
                    <td><%=item.UserPass %></td>
                    <td><%=item.RegTime %></td>
                    <td><%=item.Email %></td>
                    <td><a href='UserInf.aspx?编号=<%=item.UserID %>'>详情
                    </a></td>
                    <td><a href='Delete.aspx?编号=<%=item.UserID %>'>删除
                    </a></td>
                    <td><a href='Update.aspx?编号=<%=item.UserID %>'>修改
                    </a></td>
                </tr>
            <%}%>
        </table>
    </div>
</form>
```

# IsPostBack 分析

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="AddUserInfo.aspx.cs"
Inherits="CZBK.TestProject.WebApp._2014_11_8.aspxCURD.AddUserInfo" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
</head>
<body>
<form id="form1" runat="server">
<input type="hidden" name="isPostBack" value="1" />
<div>
<table>
<tr><td>用户名</td><td><input type="text" name="txtName" /></td></tr>
<tr><td>密码</td><td><input type="password" name="txtPwd" /></td></tr>
<tr><td>邮箱</td><td><input type="text" name="txtEmail" /></td></tr>
<tr><td colspan="3"><input type="submit" value="添加用户" /></td></tr>
</table>
</div>
</form>
</body>
</html>
```

```
public partial class AddUserInfo : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //当前用户通过 Index.aspx 页面中“添加用户”链接跳转到该页面时，这是一次 get
        请求，所以不会提交表单，拿不到隐藏域的值。当前页面显示完成，用户在表单中输入数据以后单击
        提交，这是一次 post 请求，那么这次请求会将表单中的隐藏提交到服务端。
        //if (!string.IsNullOrEmpty(Request.Form["isPostBack"]))
        if(IsPostBack)//如果你是 get 过来的那么这个值为 false, 如果是 post 过来的该为
        true.通过判断名称为__VIEWSTATE 这个隐藏域，能否获取到该隐藏域的值，来判断。
        {
            UserInfo userInfo = new UserInfo();

        }
    }
}
```
