Web Form ---IIS服务器、开发方式和简单基础

1. B/S和C/S
2. C/S

C/S架构是一种典型的两层架构，其全称是Client/Server，即客户端服务器端架构，其客户端包含一个或多个在用户的电脑上运行的程序；而服务器有两种，一种是数据库服务器端，客户端通过数据库连接访问服务器端的数据；另一种是Socket服务器端，服务器端的程序通过Socket与客户端的程序通信。

1. B/S

B/S架构全称为Browser/Server，即浏览器/服务器结构。Browser中极少数事务逻辑在前端实现，但主要事务逻辑在服务器端实现，Browser客户端，WebApp服务器端和DB端构成所谓的三层架构。

1. ASP开发模式

例子1：点菜：客户-用户

1. aspx文件

要土豆丝 – 给IIS发送请求，IIS就相当于是服务员

通知厨房 – IIS把用户想要看到的ASPX告知.NET框架

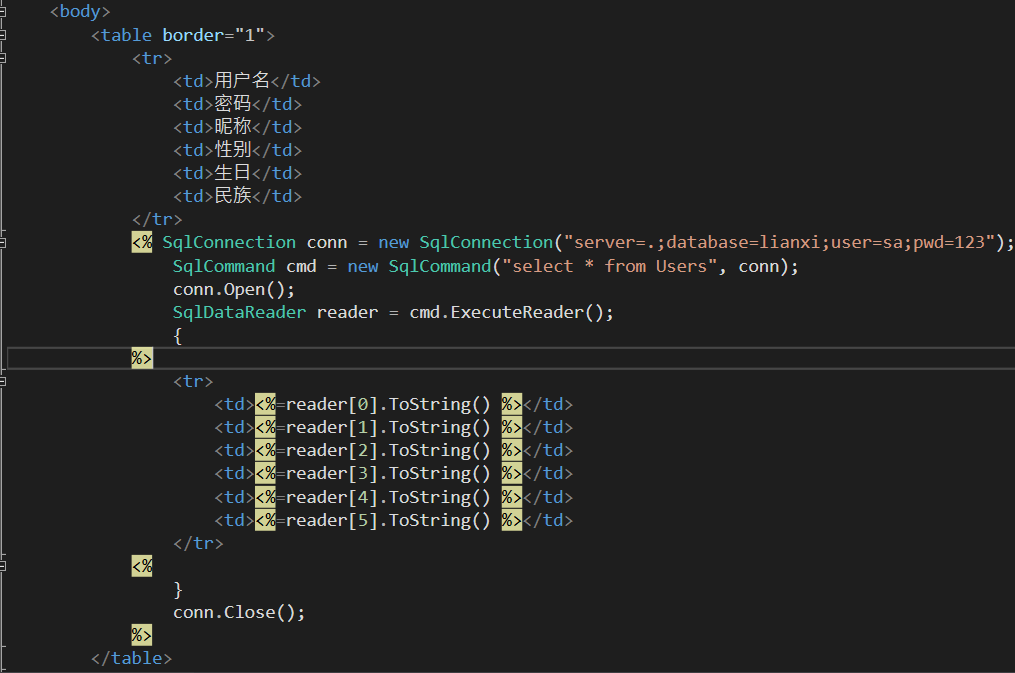
厨房炒菜 - .NET框架将ASPX页面中的C#代码执行处理

喊服务员上菜 - .NET 框架将处理好的页面给IIS，IIS返回给用户

1. html文件

要配料 – HTML页面请求，IIS直接返回给用户，不需要经过.NET 框架处理。

例子2、连接数据库，在浏览器上用表的形式显示出连接的表



<% C# 代码 %>

<%= %>等号后面返回一个 值

<%@ %>用于声明语言或者引用命名空间

<%# %>用于数据绑定，里面的表达式必须显式调用DataBind才会进行计算

1. ASP.NET

ASP.NET是.net开发网站程序的技术总称，包含WebForm和MVC 两个技术方法。

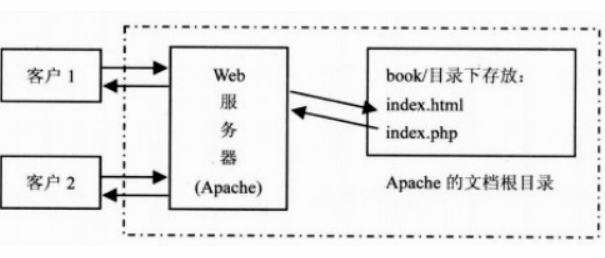
Webform类似于winform，可视化操作

MVC类似于java，多用代码操作。

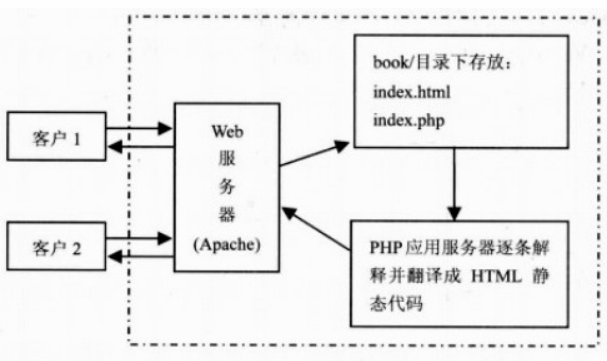
1. 空间属性

* Label – BackColor等等 最后都会变成HTML里的东西
* 控件最后都会变成HTML元素
* 界面，布局样式都依然使用html+css 特殊的数据交互需要控件来替代
* 含有runat=“server”属性的元素为控件

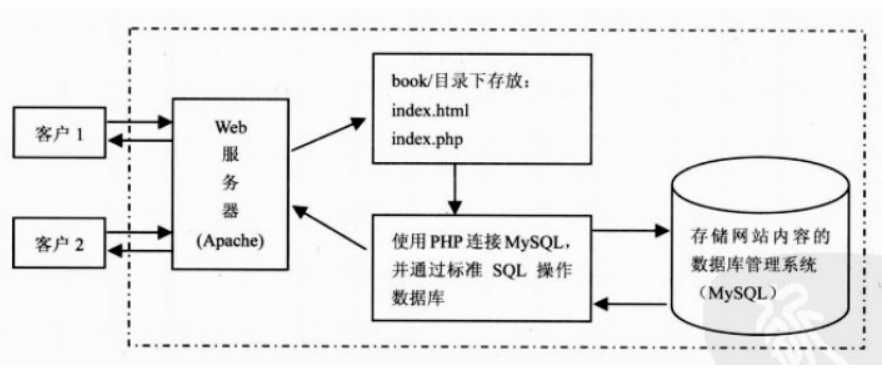
1. **三张图告诉你web工作原理**：（.net同理将图片中的.php变成.aspx文件即可）
2. 服务器不带应用程序和数据库，直接请求HTML文件

（对于一些变动不大的界面）

1. 服务器带应用程序



1. 服务器带应用程序和数据库



WebFrom---控件和数据库连接方法

1. 简单控件
2. Label（显示文字）

编译完成后的元素是span（HTML元素）

1. Literal（显示文字）

不会形成什么元素

1. TextBox（文字输入框）

TextMode：SingleLine（默认）、Password、Multiline

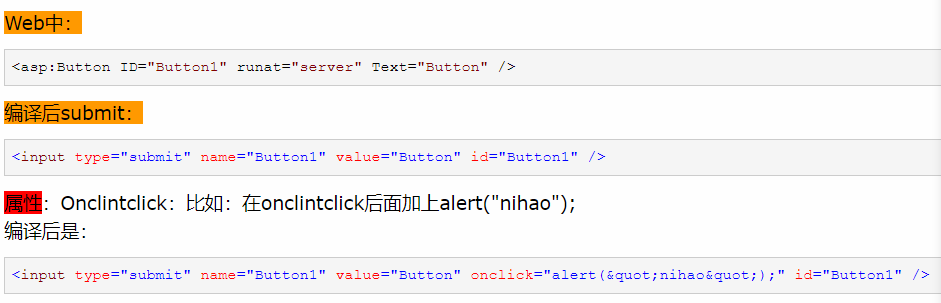
Wrap：换行（true；false）

Enable：控件启用状态

ReadOnly：是否可以更改控件中的文本

Maxlength：限制最长长度

1. Button：



上面的onclintclick应该是作者写错了。实际是onclientclick。

**注：onclientclick和onclick的区别**

Onclientclick是客户端脚本，一般使用JavaScript，在客户端运行。

Onclick是服务器端事件处理函数，使用C#，在服务器端（也就是IIS中运行），点击按钮后，执行postback，之后再运行。

Confirm()：

Confirm()方法用于显示一个带有指定消息和ok及取消按钮的对话框。

如果用户点击确定按钮，则confirm()返回true。如果点击取消按钮，则返回false。

在用户点击确定按钮或取消按钮把对话框关闭之前，他将阻止用户对浏览器的所有输入。在调用confirm时，将暂停对js代码的执行，在用户做出响应之前，不会执行下一条语句。

1. 数据库连接样式

例：做一个登录页面，连接数据库，判断是否登录成功

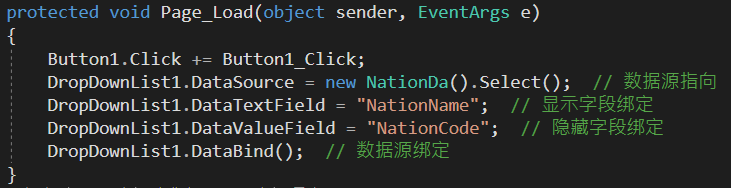
对于App\_code文件夹中的类文件，如何想要在其他文件中引用则必须将文件属性中的生产操作选项设为编译。

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="login1.aspx.cs" Inherits="Login.login1" %>

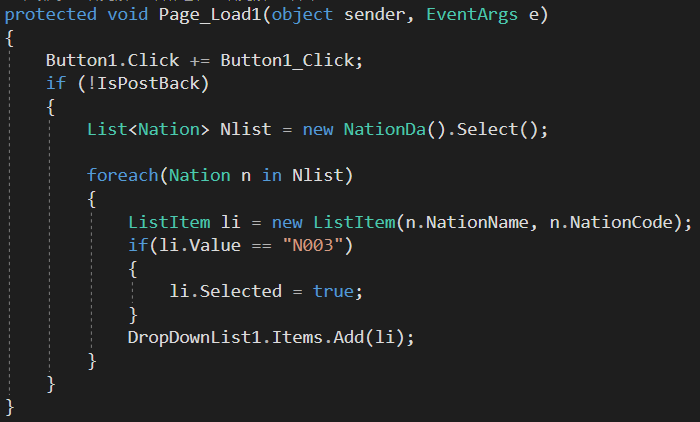
中Inherits属性中的值即对于cs文件中的class名。

1. 给DropDownList写入数据（两种方法） --放在Page\_Load中

方法一：与winform中给下拉表框添数据（DataSource）



方法二：Foreach遍历，同时加上默认选中



1. http协议无状态性：

每一次事件提交，都会将页面刷新，刷新就必走Load事件，进而重复出现绑定的情况；

解决方案：判断页面是否是第一次加载，还是由已经加载出来的页面中的某个按钮执行了提交返回回来的。

\*\*\*控件中，name用于服务端，id用于客户端。

WebForm ---三级联动

三级联动：

例：省市县三级联动

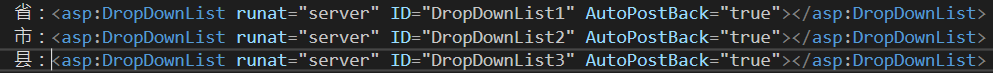
注：选第一个后面两个变，选第二个，最后一个变

其次，做三级联动需要注意的方面：

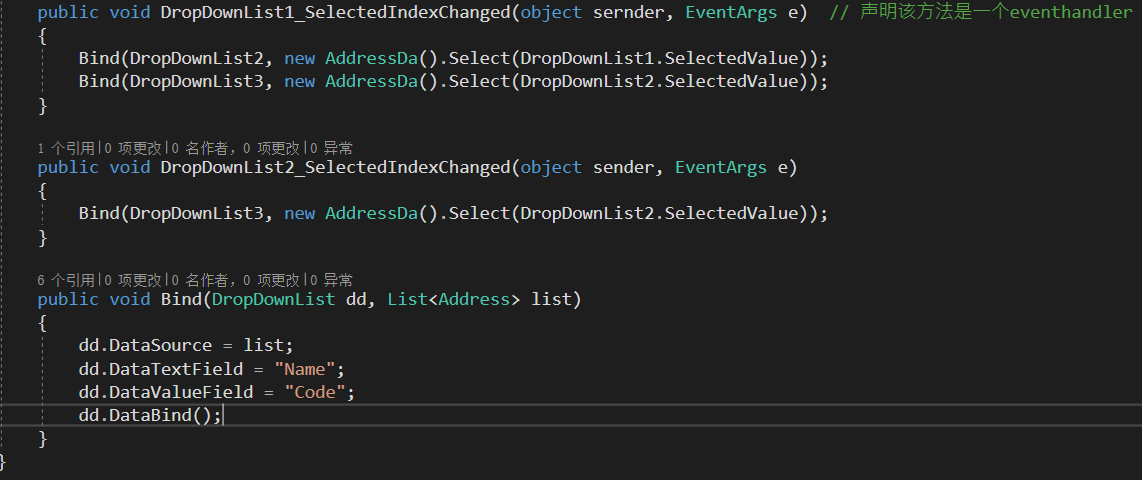
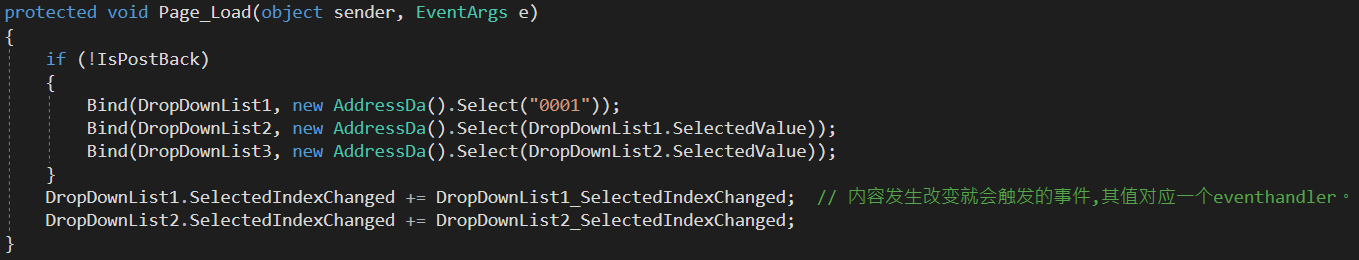
1. DropDownList中的一个属性—autopostback：是否发生自动回传到服务器的操作。如果把该属性设置为true，则启动自动回传。
2. 绑定数据（两种方法，第二种代码量多一点）

方法一：

Web中创建三个下拉列表框（DropDownList）



cs：selectIndexChanged事件：当列表控件的选定项在信息发往服务器之间变化时发生

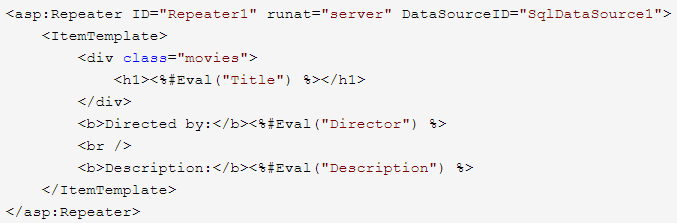


Webform—Repeater控件

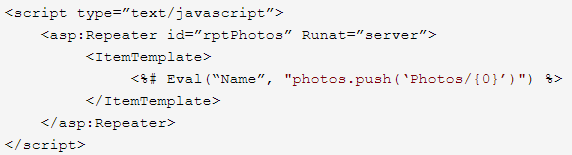
Repeater控件，可以用来一次显示一组数据项。

1. Repeater控件显示数据

要使用Repeater控件显示数据，必须创建ItemTemplate。



Repeater代码可以将其放置到JavaScript代码中。



以上，photos是一个文件夹，里面存放的是对应的图片文件

1. Repeater中使用模板

Repeater支持以下5中模板

ItemTemplate：对每一个数据项进行格式设置

AlternatingItem Template：对交替数据项进行格式设置

SeparatorTemplate：对分隔符进行格式设置

HeaderTemplate：对页眉进行格式设置，在加载开始执行一遍

FooterTemplate：对页脚进行格式设置，在加载最后执行一遍

Webform – 内置对象（Response，Request）和Repeater中的数据增删改查

1. 内置对象
2. Response对象

1、简介：response对象在ASP中负责将信息传递给用户。Response对象用于动态响应客户端请求，并将动态生成的响应结果返回给客户端浏览器中，使用response对象可以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的值等。

2、 方法：

Response.write

功能：向客户端发送浏览器能够处理的各种数据，比如：HTML代码，脚本程序等。

Redirect方法：response.redirect(“url”)的作用是在服务器端重定向到另一个网页。

1. Request对象

1、简介：Request对象的作用是与客户端交互，收集客户端的Form、Cookie、超链接，或者收集服务器端的环境变量。

2、Request对象的五个合集：

（1）QueryString：用以获取客户端附在URL地址后的查询字符串中的信息。例如：

star = Request.QueryString[“strUserId”]

前台传递的字符串：地址?key=value&key=value

注意事项：仅用于传输非重要信息

（2）Form：用以获取客户端在form表单中所输入的信息。（表单里的method属性值需要为post,get方法以字符串的格式进行传递）

例如：star = Request.form[“strUserId”]

（3）Cookies：用以获取客户端的Cookie信息。

Star = Request.Cookies[“UserId”]

（4）ServerVariables：用以获取客户端发出的HTTP请求信息中的头信息及服务器端环境变量信息。

（5）ClientCertificate：用以获取客户端的身份验证信息

例如：star = Request.ClientCertificate[“VALIDFORM”],对于要求安全验证的网站，返回有效起始日期。

1. 利用response对象和request对象对repeater中数据进行增删改查

详细见文件夹C-daily中的login。

WebForm—登录状态保持（cookie内置对象）

用户用浏览器访问一个网站，由于采用的http的特性，web服务器并不能知道是哪一个用户正在访问，但一些网站，希望能够知道访问者的一些信息，例如是不是第一次访问，访问者上次访问时是否有未做完的工作，这次是否为其继续工作提供方便等等。用浏览器访问一个网站，可以在此网站的网页之间跳转，当从第一个网页转到第二个网页时，第一个网页中建立的所有变量和对象都将不复存在有时希望在这些被访问的网页中建立联系。比如一个网上商店，访问者可能从不同的网页中选取不同的商品，那么用什么办法记录该访问者选取的商品，也就是一般所说的购物筐如何实现。用cookie对象就可以解决。

1. cookies

简单摘要：

在用户电脑的硬盘上保存一段文本

http协议包括浏览器，允许站点在用户电脑上以cookie的形式来临时保存数据

如果没有设置保存时间，即会话cookies时：

1. 如果你20分钟内没有再次刷新页面，那么cookie就会自动删除掉
2. 当前访问连接中断，如关闭浏览器，那么cookie会自动删除

如果设置保存时间，到时间后，cookie就会自动删除

作用：保持用户的登陆状态。

用法：

1. 获取账号：Response.Cookies[“user”].Value = TextBox1.Text;
2. 给账号设置登陆保持的过期时间：Response.Cookies[“user”].Expires = DateTime.Now.AddDays(7);
3. 清除cookies：Response.Cookies[“user”].Expires = DateTime.Now.AddDays(-5); 只要加上负值即可，表示已经过期几天

Webform—内置对象（session、application）和repeater的command操作

1. 内置对象
2. Session

跟Cookies一样用于存储用户数据

1. Session、Cookies对比

相同点：每一台电脑访问服务器，都会是独立的一套session或者cookies，key值都一样，但是内容都是不一样的。

不同点：Session的保存位置是保存在服务器内存上，Cookies保存在用户硬盘上

Session没有持久的，在不对页面进行任何操作的情况下，它的保存周期是20分钟；Cookies取决于用户自己设置。

1. Session重点

Session不要滥用，也不要不用，滥用可能会造成服务器内容溢出，不用会造成资源浪费。因为程序与内存之间的数据交互是最快的。

1. 用法

赋值：Session[“key”] = 值；

取值：string a = Session[“key”];

清除：Session[“key”] = null;

1. Application（全局对象）<http://www.xin3721.com/ArticlePrograme/C_biancheng/2092.html>

Application对象生存期和web应用程序生存期一样长，生存期从web应用程序网页被访问开始，HTTPApplication类对象Application被自动创建，直到没有一个网页被访问时结束，Application对象被自动撤销。因此Application对象中的变量也有相同生存期，并且变量可以被web应用程序中的所有网页访问。因此，可以在Application对象中建立一些全局的公用变量，由于存储在Application对象中的数值可以被应用程序的所有网页读取，所以Application对象的属性也适合在应用程序的网页之间传递信息。

Application对象主要有以下用途：

* 存储记录在线人数或访问网站总人数的变量。
* 存储网站共用最新消息，供所有网页更新。
* 记录网站中网页同一条广告被点击的次数或时间。
* 存储供所有网页使用的数据库数据。
* 不通用之间通讯，例如多用户聊天室，多用户游戏等。

1. 重点：

Application[“key”]是所有用户获取这个key里面的值都是一样的（通常用于获取例如版本号等通用信息的）

1. 用法：

赋值：Application[“key”] = 值

取值：Application.Get(“key”);

方法：Application对象有两个方法，它们都是用于处理多个用户对存储在Application中的数据进行写入的同步问题。由于存储在application对象中的数值可以被应用程序的所有网页读取，因此一个用户在修改这个变量时，不允许其他用户修改，这两个方法就是解决这个问题的。

1. Lock方法

Lock方法阻止其他客户修改存储在Application对象中的变量，以确保在同一时刻仅有一个客户可修改和存取Application变量。如果用户没有明确调用Unlock方法，则服务器将在.aspx文件结束或超时后即解除对Application对象的锁定。

1. Unlock方法

和Lock方法相反，Unlock方法允许其他客户修改Application对象的属性。举例如下：

Application.Lock;

Application[“counter”]=(Int32)Application[“counter”]+1;

Application.Unlock;

事件：

Application\_OnStart事件

第一个浏览器访问web应用程序网页时，产生的事件。

一般在此处设置Application的key值

<script runat=’server’>

void Application\_OnStart(object sender, EventArgs e)

{

Application.Add(“counter”, 0);

}

</script>

Application\_OnEnd事件

没有浏览器访问时web应用程序网页时，产生的事件。

Application\_OnStart和Application\_OnEnd事件的处理过程必须写在global.asax文件之中。