1. CS1061: “ASP.usercontrol\_bbsside\_ascx”不包含“loginBtn\_Click”的定义，并且找不到可接受类型为“ASP.usercontrol\_bbsside\_ascx”的第一个参数的扩展方法“loginBtn\_Click”(是否缺少 using 指令或程序集引用?)

解：这是因为在代码页中没有loginBtn\_Click这个方法。

1. nick = myDataSet.Tables["UserInfo"].Rows[0].ItemArray[0].ToString();

解：获取数据表的第一行的第一列即第一个元素的值。

还有一种方法：myDataSet.Tables["UserInfo"].Rows[0][0].ToString();

将其当做二维数组就行了。

3．我们会发现这样一个问题，当我们添加外部样式后，当我们打开前台代码时，可能什么都没有。

4. SELECT

userNick,userLevel

FROM

UserInfo

WHERE

(userPass=@Pass)and(userName=@Name)

利用sqladapter和dataset来填充数据集，返回数据集，然后再通过数据集提取数据。并不需要使用sqlcommond。

5. 运算符“<=”无法应用于“bool”和“int”类型的操作数

97 <= this.userId[0] <= 122  
  
不能这么写的，要用&&运算  
  
97 <= this.userId[0]&& this.userId[0]<= 122

6.当创建内容页后，运行内容页之后，会跟着运行应用的母版页，母版页中的web用户控件同样被运行了。Web用户控件可以根据不同的爱好进行设计，里面可以有事件属性和方法。

7．<asp:Button ID="cheknamebtn" runat="server" Text="检查用户名是否被使用" CausesValidation="False" OnClick="cheknamebtn\_Click" />

注意如果我们添加了验证控件，必须设置按钮的“是否启用验证”属性为false，不然点击按钮后会先执行验证功能再执行点击事件的功能。

7．登录验证模块

protected void loginBtn\_Click(object sender, EventArgs e)

{

//实例化一个WebService对象

localhost.Service service = new localhost.Service();

//调用webservice方法来将客户端的值传递给服务端

DataSet myDataSet;

//通过传入的用户名和密码进行登录验证

myDataSet = service.Login(nameTbx.Text.ToString().Trim(), passTbx.Text.ToString().Trim());

//记录昵称

string nick;

int level = 0;

try

{

//通过数据库传回的信息获得用户的昵称与经验值

nick = myDataSet.Tables["UserInfo"].Rows[0].ItemArray[0].ToString();//获取用户的昵称

level =Convert .ToInt32( myDataSet.Tables["UserInfo"].Rows[0].ItemArray[1]);//获取用户的等级

}

catch (Exception)

{

//登录失败用户昵称为空

nick = string.Empty;

}

//如果登录成功

if (nick != string.Empty)

{

//调用用户成功登录函数

LoginSucc(nick, nameTbx.Text.ToString(), level);

//如果用户为管理员则显示管理员面板

if (Session["username"].ToString().ToLower() == "admin")

{

adminBtn.Visible = true;

}

//登录后页面重新定向到主页面

Response.Redirect("Welcome.aspx");

return;

}

else

{

//如果登录失败则调用用户登录失败函数

LoginFal();

return;

}

}

后台服务端代码：

[WebMethod(Description = "该方法用于实现用户登录")]

public DataSet Login(string name, string pass)

{

//获得数据库连接

SqlConnection conn = new SqlConnection(DBconn);

//创建数据库操作命令

SqlCommand SqlCmd = conn.CreateCommand();

//使用pd\_Login存储过程来验证是否合法用户

SqlCmd.CommandType = CommandType.StoredProcedure;

SqlCmd.CommandText = "pd\_Login";

//传入用户名与密码参数

//参数1，用户名

SqlParameter parInput1 = SqlCmd.Parameters.Add("@Name", SqlDbType.Char);

parInput1.Direction = ParameterDirection.Input;

parInput1.Value = name.Trim();

//参数2，密码

SqlParameter parInput2 = SqlCmd.Parameters.Add("@Pass", SqlDbType.Char);

parInput2.Direction = ParameterDirection.Input;

parInput2.Value = pass.Trim();

//连接数据库

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

//执行SQL命令

SqlDataAdapter myDataAdapter = new SqlDataAdapter(SqlCmd);

DataSet myDataSet = new DataSet();

//填充数据集，并为数据集取别名

myDataAdapter.Fill(myDataSet, "UserInfo");

//获得用户等级与昵称信息

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

//返回数据集

return myDataSet;

}

登录成功后，显示经验值和用户昵称：

CREATE PROCEDURE [dbo].[pd\_Login]

(

@Name char(10),

@Pass char(10)

)

AS

SELECT

userNick,userLevel

FROM

UserInfo

WHERE

(userPass=@Pass)and(userName=@Name)

8．检查用户是否存在

public int CheckUser(string name)

{

//获得数据库连接

SqlConnection conn = new SqlConnection(DBconn);

//创建数据库操作命令

SqlCommand SqlCmd = conn.CreateCommand();

//使用pd\_checkUser存储过程来验证是否合法用户

SqlCmd.CommandType = CommandType.StoredProcedure;

SqlCmd.CommandText = "pd\_checkUser";

//传入用户名参数

SqlParameter parInput1 = SqlCmd.Parameters.Add("@Name", SqlDbType.Char);

parInput1.Direction = ParameterDirection.Input;

parInput1.Value = name.Trim();

//获得返回值

SqlParameter parReturn1 = SqlCmd.Parameters.Add("returnvalue", SqlDbType.Int);

parReturn1.Direction = ParameterDirection.ReturnValue;

//打开数据库

if (conn.State == ConnectionState.Closed)

{

conn.Open();

}

int state;

//执行Sql命令

try

{

//获得命令的返回值，获得受影响的行数

SqlCmd.ExecuteNonQuery();

//输出参数结果为0表明没有受影响，该用户不在数据库中，该参数为1表明受到影响，该用户在数据库中

state = Convert.ToInt32(SqlCmd.Parameters["returnvalue"].Value);

}

catch

{

//若查找失败 置state变量为0

state = 0;

}

if (conn.State == ConnectionState.Open)

conn.Close();

return state;

这里是存储过程：

CREATE PROCEDURE [dbo].[pd\_ChceckUser]

(

@Name char(10)

)

as

begin

select userName=@Name from UserInfo where userName=@Name

if @@Rowcount<1

return 0

else

return 1

end

8．在我们用try\_catch语句捕获异常的时候，会发现有时候我们会跳出程序返回不出我们需要的结果，这个时候我们该怎么办呢？这个时候为了测试，我们可以先不用try-catch语句使异常暴露出来，然后程序会自动告诉你是什么原因。

9.在执行存储过程时，如果我们要求输入的是两个变量则我们在试行存储过程的时候，在输入的两个变量之间一定要加一个逗号。

如--exec pd\_Login @Pass='root',@Name='admin'

10．有时候我们拷贝别人的代码的时候，尤其是前台，如果图片显示不出来怎么办，这个时候我们不妨自己拖一个控件去试一下，如果可以显示将这个路径添加到我们要显示的图片的路径就可以了。

11. Session["section"] = Request.QueryString.GetValues(0)[0].ToString();

以上代码是什么意思？

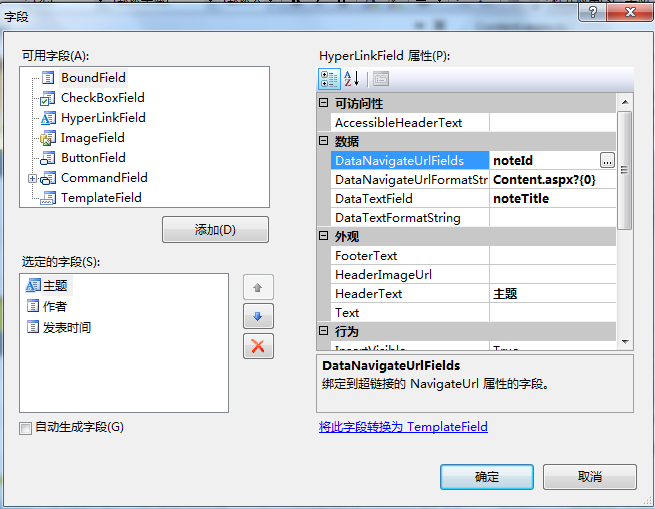
这里是获取第一个数组中的第一个元素。

答：Session的意思是会话，在web服务器端用来标识访客的变量，变量的值都是唯一的。  
而Session("变量名")表示创建一个变量 来存储这个访客的 特殊信息。  
Request是用来请求提交值的函数  
Querystring这是对应以 get方式提交的标单的 方法  
如  
<form name=标单 method=get>  
<input type=text name=section>  
<input type=submit name=ok value=提交表单>  
</form>  
Request.Querystring("section")就是获取第二行的值  
而整句话的意思就是 将这个获取的值赋给 section这个会话变量  
这样做了之后，就可以在用户访问其他的网页的时候以Session("section")这样的式子来获取section的值。

Request.QueryString {4}

为什么会出现QueryString{4}的情况呢？这是因为当前页面会使用到上一个页面的传递过来的参数，而上一页面会传递过来五组数据。这5组数据代表5个分区（站务管理、文化艺术、信息技术、休闲娱乐、学习天地），content页面会用到welcome页面的内容

注意区分以下两种情况：







由于我们在数据绑定控件datagridview中设置了编辑列，并且添加了超链接标签，在该标签中可以设置绑定到其他页面的参数

public string GetSection(string se)

{

//获得讨论分区名

switch (se)

{

case "zw":

return "站务通知";

case "wh":

return "文化艺术";

case "xx":

return "休闲娱乐";

case "td":

return "学习天地";

case "js":

return "信息技术";

}

return "";

}

protected void Page\_Load(object sender, EventArgs e)

{

string sec;

//若是页面首次被加载，获得讨论分区

if (!IsPostBack)

{

//赋值，保存在服务端作为全局变量，当浏览器关闭时session失效

//获取HTTP查询字符串变量集合

Session["section"] = Request.QueryString.GetValues(0)[0].ToString ();

}

sec = GetSection(Session["section"].ToString());

//显示论坛分区

lblName.Text = sec;

//根据不同的论坛分区显示不同的图标

IconImage();

//显示相应的论坛分区帖子列表

localhost.Service Service = new localhost.Service();

DataSet titleListDS = Service.GetSortList(sec);

GridView1.DataSource = titleListDS.Tables["TitleList"].DefaultView;

//绑定并显示留言帖标题

GridView1.DataBind();

}

//GridView换页代码

//获取或设置当前显示页的索引

GridView1.PageIndex = e.NewPageIndex;

localhost.Service Service = new localhost.Service();

//获取特定组的主题帖，获取的是当前的分区getsetction（）是方法，

DataSet titleListDS = Service.GetSortList(GetSection(Session["section"].ToString()));

//将当天分区内的所有信息显示在数据绑定控件中

GridView1.DataSource = titleListDS.Tables["TitleList"].DefaultView;

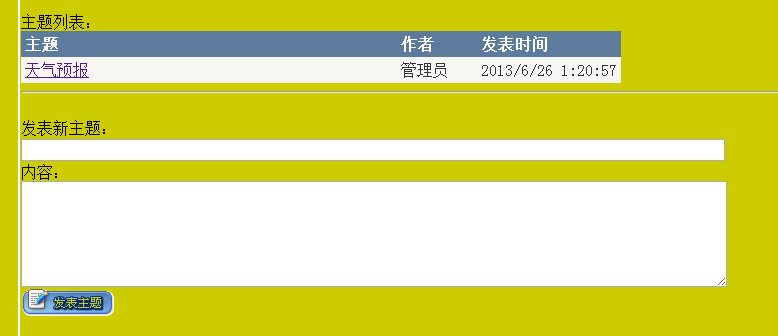
GridView1.DataBind();

欢迎界面（welcome.aspx）：



当点击“站务管理”时，直接导航到“讨论区”界面

讨论区(bbs.aspx)：



为什么我们一点击“站务管理”就能导航到“讨论区”页面呢，这是通过索引做到的，通过获取HyperLink的NavigateUrl（导航地址）来实现的。

在welcome.aspx前代码：

<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="bbs.aspx?zw">站务管理</asp:HyperLink><br />

后台bbs.aspx中添加如实代码：

//获取HTTP查询字符串变量集合，获取第一个数组中的第一个元素

Session["section"] = Request.QueryString.GetValues(0)[0].ToString ();

sec = GetSection(Session["section"].ToString());

并且通过这样的方法加以判断：

public string GetSection(string se)

{

//获得讨论分区名

switch (se)

{

case "zw":

return "站务通知";

case "wh":

return "文化艺术";

case "xx":

return "休闲娱乐";

case "td":

return "学习天地";

case "js":

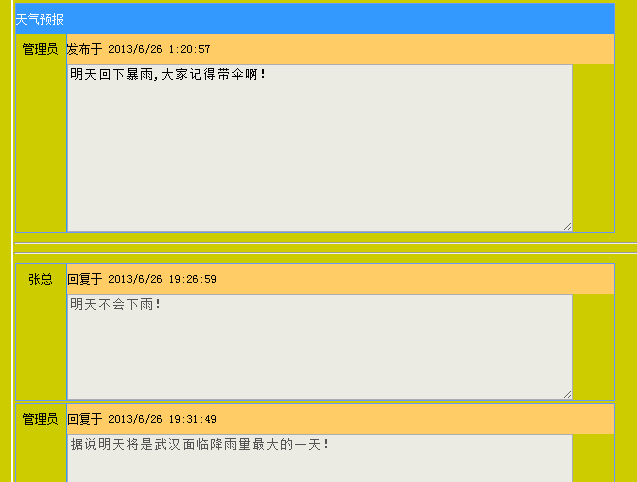
return "信息技术";

}

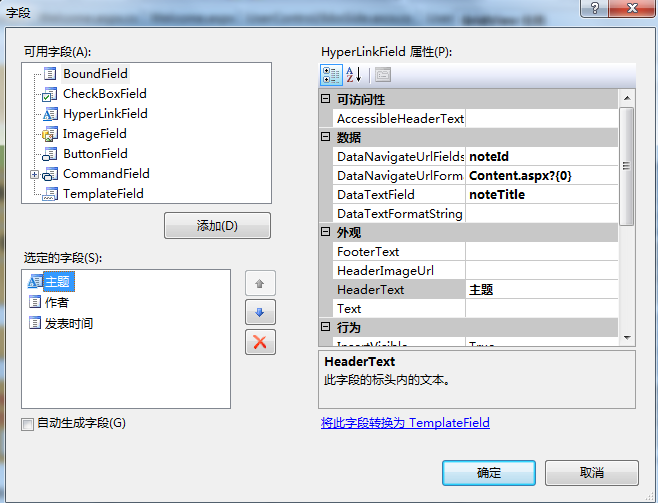
return "";

}

同理当我们点击“天气预报”，会直接导航到回复窗口，这又是怎么实现的呢？

Edit.aspx(回复窗口)  


同样是通过索引的方式（bbs窗口中gridview的编辑列属性）



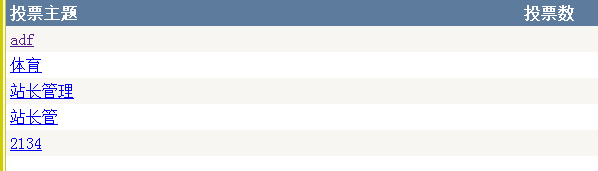
后台代码（edit.aspx）：

//显示当前主题的标题以及内容，获取id属性

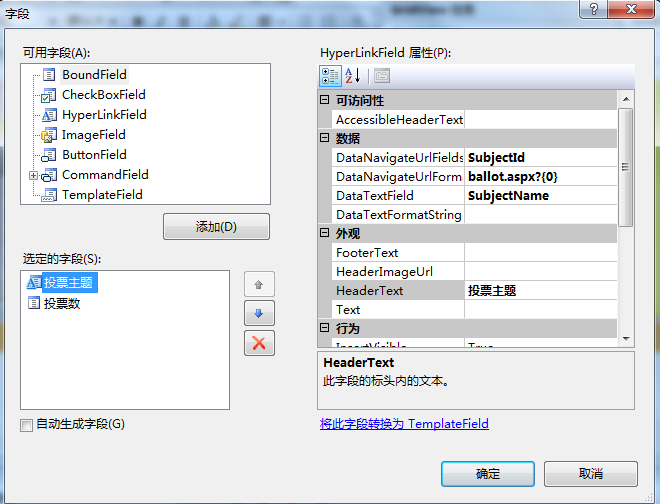
DataSet titleListDS = service.GetTitle(Convert.ToInt32(Request.QueryString.GetValues(0)[0]));

记住是在”welcome.aspx”中点击

当我们点击“在线投票”中的某一项时，会发现，会自动导航到投票页面，这是怎么做到的呢？



是通过导航



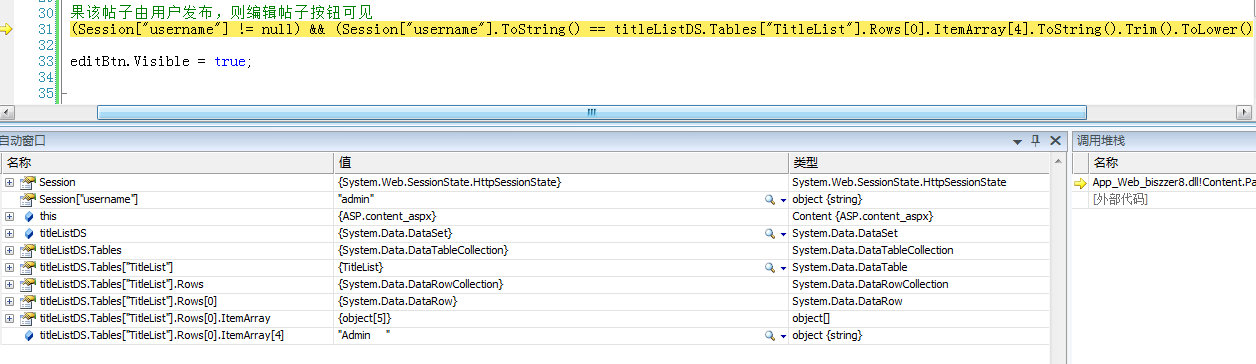
用于导航用的字段是“subjectid”

有很多投票的主题，每一个投票的主题中有很多投票的项目，比方说投票的城市等的

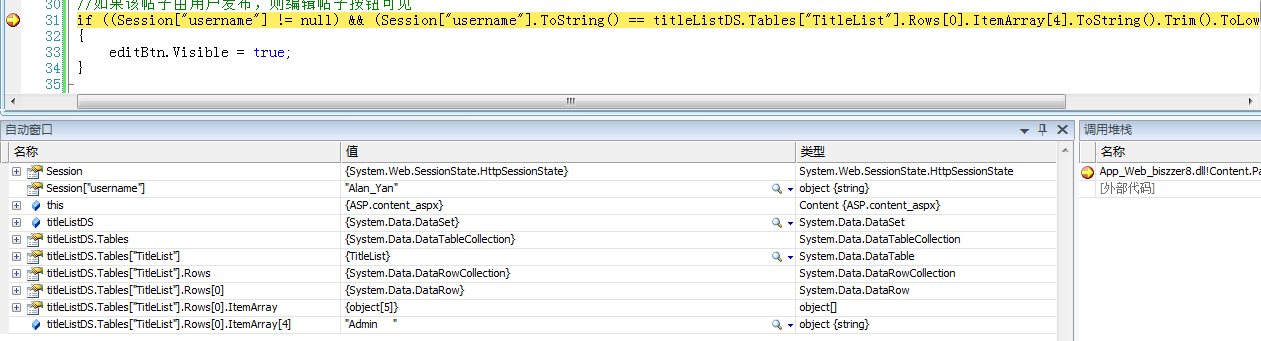
## 11. *索引超出范围。必须为非负值并小于集合大小。*

这是逻辑错误，在编系统时我们会发现当我们单独运行某一个页面时会出现索引超出范围的逻辑错误，这个错误是如何产生的呢？这是由于必须要有几个页面一起一个页面通过另一个页面传值来完成的。

12.注意区分大小写的问题



13.只有对应的发帖人才能修改自己所发帖子的内容



14.如何实现不区分大小的登录功能。

if ((Session["username"] != null) && Session["username"].ToString().Equals(titleListDS.Tables["TitleList"].Rows[0].ItemArray[4].ToString().Trim(), StringComparison.OrdinalIgnoreCase))

{

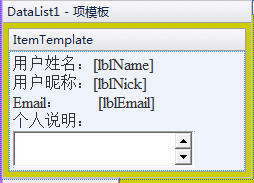
editBtn.Visible = true;

}

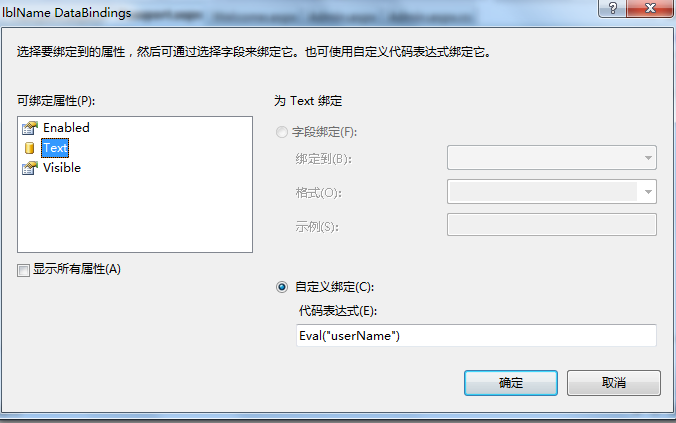
15、在datelist中我们如何实现数据绑定的功能呢？

首先，我们拖入“datelist”控件，在该控件中我们选择“编辑列”这个属性，在该属性中我们可以选择添加模板，在模板中添加一些常用控件，并且将这些控件绑定到指定的对象中。

比如，当我们想要获得某个用户的详细信息的时候，我们首先要获得该用户的全部信息，将之从数据库中读取出来并且将这些信息放入一张附表中，然后将模板中的特定的控件绑定到表中的某个元素中。



针对lblName，我们对它单独绑定元素：



底下的表达式栏即代表要绑定到的数据库中的字段名称。

16.注意：当我们在编写增加投票的模块时，我们是要设置投票的数量不为空，这样我们在做连加运算时才不至于没反应，因为空值与任何值相加都为空。