# 使用XML作为数据源

在页面中绑定xml数据源，插入、修改、删除数据

## 页面代码：

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="EditSettings1.aspx.cs" Inherits="EditSettings1" %>

<!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>Manage</title>

</head>

<body>

<form id="form1" runat="server">

<table>

<tr>

<td>Url:</td>

<td>

<asp:TextBox ID="txbUrl" runat="server" Width="260px"></asp:TextBox>

</td>

</tr>

<tr>

<td colspan="2">

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" ShowHeaderWhenEmpty="True" EmptyDataText="未添加参数" OnRowCommand="GridView1\_RowCommand" PageSize="2">

<Columns>

<asp:BoundField DataField="No" HeaderText="No" SortExpression="No" />

<asp:TemplateField HeaderText="字段名称">

<ItemTemplate>

<asp:LinkButton ID="lbtnColumn" runat="server" Text='<%#Bind("column") %>' CommandName="show" ></asp:LinkButton>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="prefix" HeaderText="前缀值" SortExpression="prefix" />

<asp:BoundField DataField="postfix" HeaderText="后缀值" SortExpression="postfix" />

<asp:TemplateField HeaderText="操作">

<ItemTemplate>

<asp:Button ID="btnDelete" runat="server" Text="删除" CommandName="del" />

<asp:Button ID="btnUp" runat="server" Text="向上" CommandName="up"/>

<asp:Button ID="btnDown" runat="server" Text="向下" CommandName="down" />

<asp:Button ID="Button1" runat="server" Text="插入" CommandName="insert"/>

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>No</td>

<td>

<asp:Label ID="Label1" runat="server" Text=""></asp:Label>

</td>

</tr>

<tr>

<td>字段名称：</td>

<td>

<asp:TextBox ID="txbColumn" runat="server" Width="250px"></asp:TextBox>

</td>

</tr>

<tr>

<td>前缀：</td>

<td>

<asp:TextBox ID="txbPrefix" runat="server" Width="250px"></asp:TextBox>

</td>

</tr>

<tr>

<td>后缀：</td>

<td>

<asp:TextBox ID="txbPostfix" runat="server" Width="250px"></asp:TextBox>

</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>

<asp:Button ID="btnSubmit" runat="server" Text="添加" OnClick="btnSubmit\_Click" />

<asp:Button ID="btnCancel" runat="server" Text="取消" OnClick="btnCancel\_Click" />

</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>

<asp:Button ID="btnSave" runat="server" Text="保存" OnClick="btnSave\_Click" />

<asp:Button ID="btnTest" runat="server" Text="test" OnClick="btnTest\_Click" />

</td>

</tr>

</table>

<asp:GridView ID="GridView2" runat="server"></asp:GridView>

</form>

</body>

</html>

## CS代码：

using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Xml.Linq;

public partial class EditSettings1 : System.Web.UI.Page

{

string Url = "taobao.com";

string xmlFilePath;

static DataTable dt;

string txtPath = @"C:\Users\Simon Fong\Desktop\新建文件夹\source code.txt";

string sourceCode;

#region 初始化GV

protected DataTable CreateDataTable()

{

DataTable dt = new DataTable();

DataColumn dc;

//添加No列

dc = new DataColumn();

dc.DataType = System.Type.GetType("System.Int32");

dc.ColumnName = "No";

dc.ReadOnly = true;

dc.Unique = true;

dt.Columns.Add(dc);

// 设置Column\_Name列为主键标识.

//DataColumn[] PrimaryKeyColumns = new DataColumn[1];

//PrimaryKeyColumns[0] = dt.Columns["No"];

//dt.PrimaryKey = PrimaryKeyColumns;

//添加Column\_Name列

dc = new DataColumn();

dc.DataType = Type.GetType("System.String");

dc.ColumnName = "column";

dc.Caption = "column";

dt.Columns.Add(dc);

//添加Prefix列

dc = new DataColumn();

dc.DataType = Type.GetType("System.String");

dc.ColumnName = "prefix";

dc.Caption = "prefix";

dt.Columns.Add(dc);

//添加Postfix列

dc = new DataColumn();

dc.DataType = Type.GetType("System.String");

dc.ColumnName = "postfix";

dc.Caption = "postfix";

dt.Columns.Add(dc);

return dt;

}

protected void GetData(string Url,string xmlFilePath)

{

XElement xe = XElement.Load(xmlFilePath);

IEnumerable<XElement> elements = from v in xe.Elements("datasource")

where v.Attribute("url").Value == Url

select v;

dt = CreateDataTable();

DataRow dr;

for (int i = 0; i < elements.Elements("column").Count(); i++)

{

dr = dt.NewRow();

dr["No"] = elements.Elements("column").ElementAt(i).Attribute("id").Value;

dr["column"] = elements.Elements("column").ElementAt(i).Attribute("name").Value;

dr["prefix"] = elements.Elements("column").ElementAt(i).Attribute("prefix").Value;

dr["postfix"] = elements.Elements("column").ElementAt(i).Attribute("postfix").Value;

dt.Rows.Add(dr);

}

}

protected void DataBindGV()

{

GridView1.DataSource = dt;

//GridView1.DataKeyNames = new string[] { "No" };

GridView1.DataBind();

}

#endregion

#region 方法

//protected bool ValidateColumn(string column)

//{

// int row\_count = dt.Rows.Count;

// string[] arr = new string[row\_count];

// for (int i = 0; i < row\_count; i++)

// {

// arr[i] = dt.Rows[i]["column"].ToString();

// }

// if (arr.Contains(column)) //如果包含column，即新Column名与已有的字段名重复

// {

// return false;

// }

// else

// {

// return true;

// }

//}

protected void Empty()

{

txbColumn.Text = txbPostfix.Text = txbPrefix.Text = "";

}

protected void InsertColumn(string column,string prefix,string postfix)

{

DataRow dr = dt.NewRow();

dr["No"] = dt.Rows.Count + 1;

dr["column"] = column;

dr["prefix"] = prefix;

dr["postfix"] = postfix;

dt.Rows.Add(dr);

}

protected void UpdateColumn(int index, string column, string prefix, string postfix)

{

dt.Rows[index][1] = column;

dt.Rows[index][2] = prefix;

dt.Rows[index][3] = postfix;

}

#endregion

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

{

xmlFilePath = Server.MapPath("App\_Data/DataCollectionSettings.xml");

if (!IsPostBack)

{

GetData(Url, xmlFilePath);

txbUrl.Text = Url;

DataBindGV();

}

}

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

{

if (btnSubmit.Text == "添加")

{

InsertColumn(txbColumn.Text.Trim(), txbPrefix.Text.Trim(), txbPostfix.Text.Trim());

DataBindGV();

Empty();

Response.Write("<script>alert('添加成功')</script>");

}

if (btnSubmit.Text == "修改")

{

UpdateColumn(Convert.ToInt32(Label1.Text)-1, txbColumn.Text.Trim(), txbPrefix.Text.Trim(), txbPostfix.Text.Trim());

DataBindGV();

Response.Write("<script>alert('修改成功')</script>");

}

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

GridViewRow gvr;

int index;

DataRow dr;

switch (e.CommandName)

{

case "show": //获取参数

gvr = (GridViewRow)(((LinkButton)(e.CommandSource)).NamingContainer);

index = gvr.RowIndex; //所在行的索引

dr = dt.Rows[index];

Label1.Text = (index+1).ToString();

txbColumn.Text = dr["column"].ToString();

txbPrefix.Text = dr["prefix"].ToString();

txbPostfix.Text = dr["postfix"].ToString();

btnSubmit.Text = "修改";

break;

case "del":

gvr = (GridViewRow)(((Button)(e.CommandSource)).NamingContainer);

index = gvr.RowIndex; //所在行的索引

dr = dt.Rows[index];

dt.Rows.RemoveAt(index);

DataBindGV();

Response.Write("<script>alert('删除成功')</script>");

break;

case "up":

gvr = (GridViewRow)(((Button)(e.CommandSource)).NamingContainer);

index = gvr.RowIndex; //所在行的索引

if (index != 0)

{

dr = dt.Rows[index];

DataRow predr = dt.Rows[index - 1];

string[] arr = new string[3];

arr[0] = predr["column"].ToString();

arr[1] = predr["prefix"].ToString();

arr[2] = predr["postfix"].ToString();

predr["column"] = dr["column"];

predr["prefix"] = dr["prefix"];

predr["postfix"] = dr["postfix"];

dr["column"] = arr[0];

dr["prefix"] = arr[0];

dr["postfix"] = arr[0];

}

DataBindGV();

break;

case "down":

gvr = (GridViewRow)(((Button)(e.CommandSource)).NamingContainer);

index = gvr.RowIndex; //所在行的索引

Label1.Text = dt.Rows.Count.ToString()+ index.ToString();

if (index < dt.Rows.Count-1)

{

dr = dt.Rows[index];

DataRow nxtdr = dt.Rows[index+ 1];

string[] arr = new string[3];

arr[0] = nxtdr["column"].ToString();

arr[1] = nxtdr["prefix"].ToString();

arr[2] = nxtdr["postfix"].ToString();

nxtdr["column"] = dr["column"];

nxtdr["prefix"] = dr["prefix"];

nxtdr["postfix"] = dr["postfix"];

dr["column"] = arr[0];

dr["prefix"] = arr[0];

dr["postfix"] = arr[0];

}

DataBindGV();

break;

}

}

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

{

Empty();

btnSubmit.Text = "添加";

}

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

{

XElement xe = XElement.Load(xmlFilePath);

IEnumerable<XElement> elementsDel = from v in xe.Elements("datasource").Elements("column")

where v.Parent.Attribute("url").Value == Url

select v;

if (elementsDel.Count() > 0)

{

elementsDel.Remove();

}

IEnumerable<XElement> elements = from v in xe.Elements("datasource")

where v.Attribute("url").Value == Url

select v;

Label1.Text = elements.Count().ToString();

foreach (DataRow dr in dt.Rows)

{

XElement newColumn = new XElement("column",

new XAttribute("id", dr["No"]),

new XAttribute("name", dr["column"]),

new XAttribute("prefix", dr["prefix"]),

new XAttribute("postfix", dr["postfix"])

);

elements.First().Add(newColumn);

}

xe.Save(xmlFilePath);

Response.Write("<script>alert('保存成功')</script>");

}

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

{

int row\_count = dt.Rows.Count;

string[,] strs = new string[row\_count, 3];

for(int i=0;i<row\_count;i++)

{

for (int j = 0; j < 3; j++)

{

strs[i, j] = dt.Rows[i][j+1].ToString();

}

}

foreach (string s in strs)

{

Response.Write(s);

}

sourceCode = Read(txtPath);

DataTable tb = CreateTable(sourceCode, strs);

int itm\_num = SubstringCount(sourceCode, strs[0, 1]);

InsertData(tb, sourceCode, strs, itm\_num);

//将数据表通过数据集绑定到grid控件

DataSet ds = new DataSet();

ds.Tables.Add(tb);

GridView2.DataSource = ds;

GridView2.DataBind();

GridView2.ShowHeaderWhenEmpty = true;

}

#region 采集数据方法

/// <summary>

/// 创建数据表

/// </summary>

/// <param name="sourcecode">源代码</param>

/// <param name="strs">设置参数（数组）</param>

/// <returns>返回带有表头的空数据表</returns>

protected DataTable CreateTable(string sourcecode, string[,] strs)

{

DataTable tb = new DataTable();

DataColumn col;

#region 创建数据表，并设置列

//创建一个int数据类型的DataColumn数据列“ID”,只读唯一，并添加到数据表tb

col = new DataColumn();

col.DataType = System.Type.GetType("System.Int32");

col.ColumnName = "ID";

col.ReadOnly = true;

col.Unique = true;

tb.Columns.Add(col);

// 设置ID列为主键标识.

DataColumn[] PrimaryKeyColumns = new DataColumn[1];

PrimaryKeyColumns[0] = tb.Columns["ID"];

tb.PrimaryKey = PrimaryKeyColumns;

//根据设置参数添加数据列

for (int colnum = 0; colnum <= strs.GetUpperBound(0); colnum++)

{

// Create second column.

col = new DataColumn();

col.DataType = Type.GetType("System.String");

col.ColumnName = strs[colnum, 0];

col.Caption = strs[colnum, 0];

tb.Columns.Add(col);

}

#endregion

return tb;

}

/// <summary>

/// 插入数据到数据表

/// </summary>

/// <param name="dt">指定的数据表</param>

/// <param name="sourcecode">源代码</param>

/// <param name="strs">设置参数（数组）</param>

/// <param name="itm\_num">重复次数</param>

protected void InsertData(DataTable dt, string sourcecode, string[,] strs, int itm\_num)

{

string \_sourcecode = sourcecode;

DataRow row;

for (int rownum = 0; rownum < itm\_num; rownum++) //生成每行数据

{

row = dt.NewRow();

row["ID"] = rownum + 1; //ID列自动累加

for (int colnum = 0; colnum <= strs.GetUpperBound(0); colnum++) //抓取数据

{

string prefix = strs[colnum, 1];

string postfix = strs[colnum, 2];

sourceCode = sourceCode.Substring(sourceCode.IndexOf(prefix) + prefix.Length);

row[colnum + 1] = sourceCode.Substring(0, sourceCode.IndexOf(postfix));

}

dt.Rows.Add(row);

}

}

/// <summary>

/// 读取指定路径的txt文件，生成字符串

/// </summary>

/// <param name="path">txt文件路径</param>

/// <returns>返回字符串</returns>

public string Read(string path)

{

FileStream fs = new FileStream(path, FileMode.Open, FileAccess.Read, FileShare.None);

StreamReader sr = new StreamReader(fs, Encoding.Default);

string txt = sr.ReadToEnd();

sr.Close();

fs.Close();

return txt;

}

/// <summary>

/// 计算目标字符串中包含指定字符串的个数

/// </summary>

/// <param name="str">目标字符串</param>

/// <param name="substring">包含的指定字符串</param>

/// <returns>返回重复出现个数</returns>

public int SubstringCount(string str, string substring)

{

if (str.Contains(substring))

{

string strRelaced = str.Replace(substring, "");

return (str.Length - strRelaced.Length) / substring.Length;

}

return 0;

}

#endregion

}