**Hướng dẫn coding gridview**

**có edit**

1. Giới thiệu:

Phần I sẽ giới thiệu các thành phần cần có để xây dựng gridview có thể edit trên lưới:



\* Nguyên tắc:

* Khai báo hai sự kiện cho gridview : onrowdatabound, onrowcommand. Dùng để bắt sự kiện khi Binding dữ liệu trên từng dòng, bắt sự kiện khi thực hiện 1 command trên 1 dòng.
* Sử dụng <asp:TemplateField></asp:TemplateField> để chứa các control. Bên trong gồm có hai thành phần cơ bản:

1. Control dùng để thể hiện dữ liệu(ví dụ asp:label)mặc định được hiện thị.
2. Các control dùng để edit, có thể nhập liệu(ví dụ asp:TextBox, asp:Combobox,…) chỉ được hiện thị khi có 1 command trên row(row được edit)

* Trong column đầu tiên phải có thành phần:

<asp:ButtonField Text="SingleClick" CommandName="SingleClick" Visible="False" />. Dùng kiểm soát sự kiện khi click trên row.

1. Cách sử dụng
2. Các thuộc tính cần phải khai báo:
3. private int \_iTabIndex = 0; Dùng chứa tabIndex khi nhấn phím Tab.
4. <asp:HiddenField ID="hidReRow1" Value="-1" runat="server"/>: Dùng để chứa row có command mỗi lần sự kiện onrowcommand,
5. <asp:HiddenField ID="hidGridStatus1" Value="-1" runat="server"/>: Dùng đế chứa tình trạng của lưới. Không có dòng nào: Value =”empty”, ngược lại chứa số dòng của lưới trên database

1. Các hàm chính trong code behind
2. protected override void Render(HtmlTextWriter writer){}: đang kí script cho client trên dòng và cột của lưới

Ví dụ:

protected override void Render(HtmlTextWriter writer)

{

foreach (GridViewRow Rows in grdExample.Rows)

{

if (Rows.RowType == DataControlRowType.DataRow)

{

for (int columnIndex = 2; columnIndex < Rows.Cells.Count; columnIndex++)

{

Page.ClientScript.RegisterForEventValidation(Rows.UniqueID + "$ctl00", columnIndex.ToString());

}

}

}

base.Render(writer);

}

1. private void \_AddNewRowGrid(int Row){}: thêm một dòng mới vào lưới, không có dữ liệu dùng để insert dòng mới. Có ba trường hợp phải add dòng mới vào lưới:
   1. Row == grdExample.Rows.Count – 1:dòng hiện tại đang đứng là dòng cuối.
   2. Row == -1:Trường hợp load đâu tiên
   3. hidGridStatus1.Value == "empty": trường hợp csdl không có dòng nào
2. protected void grdExample\_RowDataBound(object sender, GridViewRowEventArgs e): bắt sự kiện bound dữ liệu trên dòng.

Trong hàm này sẽ tìm tất cả các label của dòng hiện tại và gắn dữ liệu từ datasource:

PUBCURRENCYRATE curr = (PUBCURRENCYRATE)e.Row.DataItem;

if (e.Row.RowType == DataControlRowType.DataRow)

{

GridViewRow Row = (GridViewRow)e.Row;

// Tìm tất cả label trên dòng

List<Label> \_lstLabel = new List<Label>();

\_lstLabel.Add((Label)Row.FindControl("lblIndex"));

\_lstLabel.Add((Label)Row.FindControl("lblEntryMoney"));

\_lstLabel.Add((Label)Row.FindControl("lblDate"));

\_lstLabel.Add((Label)Row.FindControl("lblSourceRate"));

\_lstLabel.Add((Label)Row.FindControl("lblAffectiveDate"));

\_lstLabel.Add((Label)Row.FindControl("lblDescription"));

//Gán dữ liệu vào label

\_lstLabel[0].Text = e.Row.RowIndex.ToString();

\_lstLabel[1].Text = curr.CRR\_ENTRYCURRENCYID.ToString();

\_lstLabel[2].Text = curr.CRR\_DATE.ToString();

\_lstLabel[3].Text = curr.CRR\_SOURCERATE;

\_lstLabel[4].Text = curr.CRR\_ACTIVEDATE.ToString();

\_lstLabel[5].Text = curr.CRR\_DESCRIPTION;

//Gán tabindex cho label

foreach (Label lb in \_lstLabel)

{

lb.TabIndex = Convert.ToInt16(\_iTabIndex++);

}

// và đăng kí sự kiện onclick cho row, và focus cho label:

LinkButton \_singleClickButton = (LinkButton)e.Row.Cells[0].Controls[0];

string \_jsSingle = ClientScript.GetPostBackClientHyperlink(\_singleClickButton, "");

for (int columnIndex = 2; columnIndex < e.Row.Cells.Count; columnIndex++)

{

string js = \_jsSingle.Insert(\_jsSingle.Length - 2, columnIndex.ToString());

//Đăng kí focus cho label

foreach (Label lb in \_lstLabel)

{

lb.Attributes["onfocus"] = js;

}

//Đăng kí sự kiên onclik trên row

Row.Cells[columnIndex].Attributes["onclick"] = js;

e.Row.Cells[columnIndex].Attributes["style"] += "cursor:pointer;cursor:hand;";

}

}

}

1. protected void grdExample\_RowCommand(object sender, GridViewCommandEventArgs e): Bắt sự kiện trên dòng(click).

Khai báo hai biết cục bộ là activerow: index của dòng hiện tại và reindex: index của dòng lần click trước. Cần qua các bước:

1. Nếu là sự kiện sigleclick
2. Kiển tra xem indexrow<>reindex:
   1. Nếu khác:
      * + Ẩn toàn bộ control nhập dữ liệu dòng reindex của lưới
        + Hiện toàn bộ control label dòng reindex của lưới
        + Gán tabindex cho control nhập liệu
        + Gán dự liệu từ các control nhập liệu của dong reindex vào đối tượng và kiểm tra cho insert or update.
        + Gán sự kiện cho cell
   2. Ngược lại: không làm gì
3. Đối với dòng activerow: ẩn toàn bộ control label, hiện toàn bộ control nhập liệu. Clear toàn bộ sự kiện trên cell
4. Gán lại reindex = activerow

Chi tiết xem ví dụ

1. Nếu sự kiện là delete

Xóa dòng hiện tại của lưới và trong csdl tương ứng.

1. Ví dụ:
2. Trên trang aspx
3. <div style="overflow:scroll; width:700px; height:inherit; height:500px">
4. <asp:HiddenField ID="hidGridStatus1" Value="-1" runat="server"/>
5. <asp:HiddenField ID="hidReRow1" Value="-1" runat="server"/>
7. <asp:GridView ID="grdCategorie" runat="server" AutoGenerateColumns="False"
8. Width="650px" BorderStyle="Solid" BorderWidth="1px"
9. onrowcommand="grdCategorie\_RowCommand"
10. onrowdatabound="*grdCategorie*\_RowDataBound" >
11. <Columns>
12. <asp:ButtonField Text="SingleClick" CommandName="SingleClick" Visible="False" />
13. <asp:TemplateField HeaderText="STT" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
14. <ItemTemplate>
15. <asp:Label ID="lblIndex" runat="server" ></asp:Label>

<asp:Label ID="lblID" runat="server" ></asp:Label>

1. </ItemTemplate>
2. </asp:TemplateField>
3. <asp:TemplateField HeaderText="Tiền hoạch toán" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
4. <ItemTemplate>
5. <asp:Label ID="lblEntryMoney" runat="server" ></asp:Label>
6. <asp:TextBox ID="txtEntryMoneyID" runat="server" Visible="false"></asp:TextBox>
7. <dxe:ASPxButtonEdit ID="txtEntryMoney" runat="server" Visible ="false" >
8. <Buttons>
9. <dxe:EditButton >
11. </dxe:EditButton>
13. </Buttons>
14. <ClientSideEvents GotFocus ="function(s, e) {
15. showDialog();
16. }" ButtonClick="function(s, e) {
18. showDialog();
19. }" />
20. </dxe:ASPxButtonEdit>
21. <asp:HiddenField ID="RownID" runat="server" />
22. </ItemTemplate>
23. </asp:TemplateField>
24. <asp:TemplateField HeaderText="Ngày tỷ giá" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
25. <ItemTemplate>
26. <asp:Label ID="lblDate" runat="server" ></asp:Label>
27. <dxe:ASPxDateEdit ID="txtDate" runat="server" Visible ="false">
28. </dxe:ASPxDateEdit>
30. </ItemTemplate>
31. </asp:TemplateField>
32. <asp:TemplateField HeaderText="Nguồn tỷ giá" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
33. <ItemTemplate>
34. <asp:Label ID="lblSourceRate" runat="server" ></asp:Label>
35. <dxe:ASPxButtonEdit ID="txtSourceRate" runat="server" Visible ="false">
36. <Buttons>
37. <dxe:EditButton>
38. </dxe:EditButton>
39. </Buttons>
40. </dxe:ASPxButtonEdit>
41. </ItemTemplate>
42. </asp:TemplateField>
43. <asp:TemplateField HeaderText="Ngày hiệu lực" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
44. <ItemTemplate>
45. <asp:Label ID="lblAffectiveDate" runat="server" ></asp:Label>
47. <dxe:ASPxDateEdit ID="txtAffectiveDate" runat="server" Visible ="false">
48. </dxe:ASPxDateEdit>
49. </ItemTemplate>
50. </asp:TemplateField>
51. <asp:TemplateField HeaderText="Diễn giải" ItemStyle-HorizontalAlign="Center" ItemStyle-VerticalAlign="Middle" >
52. <ItemTemplate>
53. <asp:Label ID="lblDescription" runat="server" ></asp:Label>
54. <asp:TextBox ID="txtDescription" runat="server" Visible="false"></asp:TextBox>
55. </ItemTemplate>
56. </asp:TemplateField>
57. <asp:ButtonField ButtonType="Button" Text="Delete"/>
58. </Columns>
59. <AlternatingRowStyle BackColor="#FBDC89" />
60. <HeaderStyle BackColor="#ECE9D8" />
61. </asp:GridView>
62. </div>
63. Trên code behind

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using vnyi.Accounting;

using vnyi.Utility;

using DevExpress.Web.ASPxEditors;

using DevExpress.Web.ASPxGridView;

namespace vnyi.EziPos

{

public partial class CurrencyRate\_ : System.Web.UI.Page

{

private int \_iTabIndex = 0;

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

{

if (!IsPostBack)

\_LoadGridCategorie();

}

protected override void Render(HtmlTextWriter writer)

{

foreach (GridViewRow Rows in grdCategorie.Rows)

{

if (Rows.RowType == DataControlRowType.DataRow)

{

for (int columnIndex = 2; columnIndex < Rows.Cells.Count; columnIndex++)

{

Page.ClientScript.RegisterForEventValidation(Rows.UniqueID + "$ctl00", columnIndex.ToString());

}

}

}

base.Render(writer);

}

/// <summary>

/// Khoi tao luoi

/// </summary>

private void \_LoadGridCategorie()

{

List<PUBCURRENCYRATE> lstCurr =PUBCURRENCYRATEDAO.GetAllPUBCURRENCYRATE();

PUBCURRENCYRATE curr;

if (grdCategorie.DataSource != null)

{

\_AddNewRowGridCategorie(-2);

}

if (grdCategorie.Rows.Count == 1)

hidGridStatus1.Value = "empty";

else if (grdCategorie.Rows.Count == 0)

{

curr = new PUBCURRENCYRATE();

curr.CRR\_ACTIVEDATE = DateTime.Now;

curr.CRR\_DATE = DateTime.Now;

curr.CRR\_DESCRIPTION = "Emtry";

curr.CRR\_ENTRYCURRENCYID = 0;

curr.CRR\_RATE = 0;

hidGridStatus1.Value = "empty";

lstCurr.Add(curr);

}

grdCategorie.DataSource =lstCurr;

grdCategorie.DataBind();

}

/// <summary>

/// them dong moi vo luoi

/// </summary>

private void \_AddNewRowGridCategorie(int count)

{

try

{

List<PUBCURRENCYRATE> lstCurr = PUBCURRENCYRATEDAO.GetAllPUBCURRENCYRATE();

PUBCURRENCYRATE curr;

if (count == grdCategorie.Rows.Count - 1)

{

curr = new PUBCURRENCYRATE();

curr.CRR\_ACTIVEDATE = DateTime.Now;

curr.CRR\_DATE = DateTime.Now;

curr.CRR\_DESCRIPTION = "Emtry";

curr.CRR\_ENTRYCURRENCYID =0;

curr.CRR\_RATE = 0;

lstCurr.Add(curr);

}

else if (count == -1)

{

curr = new PUBCURRENCYRATE();

curr.CRR\_ACTIVEDATE = DateTime.Now;

curr.CRR\_DATE = DateTime.Now;

curr.CRR\_DESCRIPTION = "Emtry";

curr.CRR\_ENTRYCURRENCYID = 0;

curr.CRR\_RATE = 0;

lstCurr.Add(curr);

}

if (hidGridStatus1.Value == "empty")

{

curr = new PUBCURRENCYRATE();

curr.CRR\_ACTIVEDATE = DateTime.Now;

curr.CRR\_DATE = DateTime.Now;

curr.CRR\_DESCRIPTION = "Emtry";

curr.CRR\_ENTRYCURRENCYID = 0;

curr.CRR\_RATE = 0;

lstCurr.Add(curr);

}

grdCategorie.DataSource = lstCurr;

grdCategorie.DataBind();

}

catch (Exception ex)

{ }

}

/// <summary>

/// do du lieu vao luoi

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

protected void grdCategorie\_RowDataBound(object sender, GridViewRowEventArgs e)

{

PUBCURRENCYRATE curr = (PUBCURRENCYRATE)e.Row.DataItem;

if (e.Row.RowType == DataControlRowType.DataRow)

{

GridViewRow Row = (GridViewRow)e.Row;

List<Label> \_lstLabel = new List<Label>();

\_lstLabel.Add((Label)Row.FindControl("lblIndex"));

\_lstLabel.Add((Label)Row.FindControl("lblEntryMoney"));

\_lstLabel.Add((Label)Row.FindControl("lblDate"));

\_lstLabel.Add((Label)Row.FindControl("lblSourceRate"));

\_lstLabel.Add((Label)Row.FindControl("lblAffectiveDate"));

\_lstLabel.Add((Label)Row.FindControl("lblDescription"));

\_lstLabel[0].Text = e.Row.RowIndex.ToString();

\_lstLabel[1].Text = curr.CRR\_ENTRYCURRENCYID.ToString();

\_lstLabel[2].Text = curr.CRR\_DATE.ToString();

\_lstLabel[3].Text = curr.CRR\_SOURCERATE;

\_lstLabel[4].Text = curr.CRR\_ACTIVEDATE.ToString();

\_lstLabel[5].Text = curr.CRR\_DESCRIPTION;

foreach (Label lb in \_lstLabel)

{

lb.TabIndex = Convert.ToInt16(\_iTabIndex++);

}

LinkButton \_singleClickButton = (LinkButton)e.Row.Cells[0].Controls[0];

string \_jsSingle = ClientScript.GetPostBackClientHyperlink(\_singleClickButton, "");

for (int columnIndex = 2; columnIndex < e.Row.Cells.Count; columnIndex++)

{

string js = \_jsSingle.Insert(\_jsSingle.Length - 2, columnIndex.ToString());

foreach (Label lb in \_lstLabel)

{

lb.Attributes["onfocus"] = js;

}

Row.Cells[columnIndex].Attributes["onclick"] = js;

e.Row.Cells[columnIndex].Attributes["style"] += "cursor:pointer;cursor:hand;";

}

}

}

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

{

GridView \_gridView = (GridView)sender;

switch (e.CommandName)

{

case ("SingleClick"):

// Lấy thông tin của lưới

int \_reIndex = int.Parse(hidReRow1.Value);

int \_rowIndex = int.Parse(e.CommandArgument.ToString());

// Sau hai lần postback thứ hai nếu lần click thứ 2 khác row lần click trước đó

if (\_rowIndex != \_reIndex)

{

//Nếu dòng click lần trước thuộc về lưới

if (\_reIndex >= 0)

{

// Tìm kiếm các lable và các control dùng để nhập dữ liệu

GridViewRow RowRe = \_gridView.Rows[\_reIndex];

hidReRow1.Value = \_rowIndex.ToString();

List<Label> \_lstLabel = new List<Label>();

\_lstLabel.Add((Label)RowRe.FindControl("lblIndex"));

\_lstLabel.Add((Label)RowRe.FindControl("lblEntryMoney"));

\_lstLabel.Add((Label)RowRe.FindControl("lblDate"));

\_lstLabel.Add((Label)RowRe.FindControl("lblSourceRate"));

\_lstLabel.Add((Label)RowRe.FindControl("lblAffectiveDate"));

\_lstLabel.Add((Label)RowRe.FindControl("lblDescription"));

TextBox txtEntryMoneyID = (TextBox)RowRe.FindControl("txtEntryMoneyID");

List<object> objs = new List<object>();

objs.Add((TextBox)RowRe.FindControl("txtIndex"));

objs.Add((DevExpress.Web.ASPxEditors.ASPxButtonEdit)RowRe.FindControl("txtEntryMoney"));

objs.Add((DevExpress.Web.ASPxEditors.ASPxDateEdit)RowRe.FindControl("txtDate"));

objs.Add((DevExpress.Web.ASPxEditors.ASPxButtonEdit)RowRe.FindControl("txtSourceRate"));

objs.Add((DevExpress.Web.ASPxEditors.ASPxDateEdit)RowRe.FindControl("txtAffectiveDate"));

objs.Add((TextBox)RowRe.FindControl("txtDescription"));

//gan tabindex

for (int i = 1; i < \_lstLabel.Count; i++)

{

((Control)objs[i]).Visible = false;

\_lstLabel[i].Visible = true;

}

((Control)objs[1]).Focus();

//lấy dự liệu được nhập vào

PUBCURRENCYRATE curr = new PUBCURRENCYRATE();

curr.CRR\_ACTIVEDATE = ((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs[4]).Date;

curr.CRR\_CURRENCYID = clsFormat.Int16Convert(txtEntryMoneyID.Text);

curr.CRR\_DATE = ((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs[2]).Date;

curr.CRR\_DESCRIPTION = ((TextBox)objs[5]).Text;

curr.CRR\_ENTRYCURRENCYID = clsFormat.Int16Convert(txtEntryMoneyID.Text);

curr.CRR\_RATE = 1;

curr.CRR\_SOURCERATE = ((DevExpress.Web.ASPxEditors.ASPxButtonEdit)objs[3]).Text;

//Kiểm tra trường hợp cho insert hay update

if (hidGridStatus1.Value == "empty" || \_rowIndex == \_gridView.Rows.Count - 1)

{

hidGridStatus1.Value = \_gridView.Rows.Count.ToString();

PUBCURRENCYRATEDAO.Insert(curr);

}

else

{

PUBCURRENCYRATEDAO.Update(curr);

}

// gắn sự kiện cho cell

LinkButton \_singleClickButton = (LinkButton)RowRe.Cells[0].Controls[0];

string \_jsSingle = ClientScript.GetPostBackClientHyperlink(\_singleClickButton, "");

for (int columnIndex = 2; columnIndex < RowRe.Cells.Count; columnIndex++)

{

string js = \_jsSingle.Insert(\_jsSingle.Length - 2, columnIndex.ToString());

RowRe.Cells[columnIndex].Attributes["onclick"] = js;

//e.Row.Cells[columnIndex].Attributes["onclick"] = js;

RowRe.Cells[columnIndex].Attributes["style"] += "cursor:pointer;cursor:hand;";

}

}

//hien thi du lieu

if (\_rowIndex >= \_gridView.Rows.Count - 1)

{

//them dong moi

\_AddNewRowGridCategorie(\_rowIndex);

}

GridViewRow Row = \_gridView.Rows[\_rowIndex];

hidReRow1.Value = \_rowIndex.ToString();

List<Label> labels = new List<Label>();

labels.Add((Label)Row.FindControl("lblIndex"));

labels.Add((Label)Row.FindControl("lblEntryMoney"));

labels.Add((Label)Row.FindControl("lblDate"));

labels.Add((Label)Row.FindControl("lblSourceRate"));

labels.Add((Label)Row.FindControl("lblAffectiveDate"));

labels.Add((Label)Row.FindControl("lblDescription"));

TextBox txtEntryMoneyID2 = (TextBox)Row.FindControl("txtEntryMoneyID");

List<object> objs2 = new List<object>();

objs2.Add((TextBox)Row.FindControl("txtIndex"));

objs2.Add((DevExpress.Web.ASPxEditors.ASPxButtonEdit)Row.FindControl("txtEntryMoney"));

objs2.Add((DevExpress.Web.ASPxEditors.ASPxDateEdit)Row.FindControl("txtDate"));

objs2.Add((DevExpress.Web.ASPxEditors.ASPxButtonEdit)Row.FindControl("txtSourceRate"));

objs2.Add((DevExpress.Web.ASPxEditors.ASPxDateEdit)Row.FindControl("txtAffectiveDate"));

objs2.Add((TextBox)Row.FindControl("txtDescription"));

//gan tabindex

for (int i = 1; i < labels.Count; i++)

{

//hiện các control nhập dữ liệu và ẩn các control label cho dòng được click

((Control)objs2[i]).Visible = true;

labels[i].Visible = false;

}

((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs2[4]).Date = clsFormat.DateConvert(labels[4].Text);

((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs2[4]).TabIndex = labels[4].TabIndex;

((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs2[2]).Date = clsFormat.DateConvert(labels[2].Text); ;

((DevExpress.Web.ASPxEditors.ASPxDateEdit)objs2[2]).TabIndex = labels[2].TabIndex;

((DevExpress.Web.ASPxEditors.ASPxButtonEdit)objs2[1]).Text = labels[1].Text;

((DevExpress.Web.ASPxEditors.ASPxButtonEdit)objs2[1]).TabIndex = labels[1].TabIndex;

((DevExpress.Web.ASPxEditors.ASPxButtonEdit)objs2[3]).Text = labels[3].Text;

((DevExpress.Web.ASPxEditors.ASPxButtonEdit)objs2[3]).TabIndex = labels[3].TabIndex;

((TextBox)objs2[5]).Text = labels[5].Text;

((TextBox)objs2[5]).TabIndex = labels[5].TabIndex;

((Control)objs2[1]).Focus();

//Xóa bỏ các sự kiện các cell của dòng hiện tại. Tráng trường hợp postback liên tục

for (int columnIndex = 2; columnIndex < Row.Cells.Count; columnIndex++)

{

Row.Cells[columnIndex].Attributes.Clear();

}

}

hidReRow1.Value = \_rowIndex.ToString();

break;

case ("Delete"):

GridViewRow Rowd = \_gridView.Rows[\_rowIndex];

Label lblID = (Label)Rowd.FindControl("lblIndex"));

PUBCURRENCYRATEDAO.Delete(clsFormat.IntConvert( lblID.Text));

break;

}

}

}

}