

IN FOCUS



ASK A QUESTION

CONTRIBUTE

ARTICLE

READER LEVEL:



Select, Add, Update, and Delete Data in a ASP.NET GridView Control

By [Raj Kumar](#) on Nov 23, 2008

This articles describes you how to view, add, update, and delete data using an ASP.NET 2.0 GridView Control in SQL Server database and C#.

230.4 k

40

5

Download Files: [GridViewExample.zip](#)[Download 100% FREE Spire Office APIs](#)

In my previous article, I described how to use a GridView control to display data using a SqlDataSource control. In this article, I will show you how to use an ASP.NET 2.0 GridView control to view, add, update, and delete data in a SQL database.

We will use SQL Client data provider to provide database connectivity.

Before you can use any classes related to SQL Client data adapter, we need to import the SqlClient namespace in your application by using the following using statement.

```
using System.Data.SqlClient;
```

Next, we need to define the database connection string.

The below is my connection string which is stored in web.config file. You can change this connection string according to your SQL server database setting. I am storing my database file in App_Data folder. If you want use my database file then attach that file.

```
<appSettings>
<add key="ConnectionString" value="Data Source=.;Initial Catalog=Vendor;User
ID=sa;Password=wintellect;" />
</appSettings>
```

The following code snippet shows how to connect to a database and create other database access related objects.

```
SqlConnection conn = new
SqlConnection(ConfigurationManager.AppSettings["ConnectionString"].ToString());
SqlDataAdapter ad = new SqlDataAdapter();
SqlCommand cmd = new SqlCommand();
DataTable dataTable;
```

This function is use to fetch data from the Vendor table, fills data in a DataTable object and find it to a GridView control using the DataSource property. In the end, the code calls the GridView.DataBind method to apply the binding.

```
private void FillVendorGrid()
```

```

{
    dataTable = new DataTable();
    cmd.Connection = conn;
    cmd.CommandText = "SELECT * FROM Vendor";
    ad = new SqlDataAdapter(cmd);
    ad.Fill(dataTable);
    ResultGridView.DataSource = dataTable;
    ResultGridView.DataBind();
}

```

Now on the page load method, we call the FillVendorGrid method.

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        FillVendorGrid();
    }
}

```

Now, next step is to set the GridView control settings.

The ASP.NET code for the DataView control. In this code below code, you see database table columns binding with the bound fields and formatting is provided using the template fields. If you are using my database, just copy and paste the code or use the attached application. If you are using your database, you need to replace column binding with your database table columns.

```

<asp:GridView ID="ResultGridView" runat="server" AutoGenerateColumns="False"
ShowFooter="true"
DataKeyNames="VendorId"
AllowPaging="True"
CellPadding="3"
OnPageIndexChanging="ResultGridView_PageIndexChanging"
OnRowDeleting="ResultGridView_RowDeleting"
OnRowEditing="ResultGridView_RowEditing"
OnRowUpdating="ResultGridView_RowUpdating"
OnRowCancelingEdit="ResultGridView_RowCancelingEdit"
PageSize="5"
BackColor="#DEBA84"
BorderColor="#DEBA84"
BorderStyle="None"
BorderWidth="1px"
CellSpacing="2"
OnRowCommand="ResultGridView_RowCommand">
<Columns>
<asp:BoundField DataField="VendorId" HeaderText="VendorId" InsertVisible="False"
ReadOnly="True" SortExpression="VendorId" />
<asp:TemplateField HeaderText="FirstName" SortExpression="VendorFName">

```

C# Corner Discontinues The Code Snippet Section

[TECHNOLOGIES](#)
[ANSWERS](#)
[BLOGS](#)
[VIDEOS](#)
[INTERVIEWS](#)
[BOOKS](#)
[C# Corner Search](#)
[NEWS](#)
[CHAPTERS](#)
[CAREER](#)
[JOBS](#)
[IDEAS](#)

```

r" Text='<%# Bind("VendorFName") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="LastName" SortExpression="VendorLName">
<EditItemTemplate>
<asp:TextBox ID="txtLName" Width="100px" runat="server" Text='<%#

```

```

</ItemTemplate>
</asp:TemplateField>

```

```

</asp:TemplateField>

```

```

<asp:TemplateField HeaderText="LastName" SortExpression="VendorLName">

```

```

<EditItemTemplate>

```

```

<asp:TextBox ID="txtLName" Width="100px" runat="server" Text='<%#

```

```

Bind("VendorLName") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox ID="txtLName1" Width="100px" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Label2" runat="server" Text='<%= Bind("VendorLName") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="City" SortExpression="VendorCity">
<EditItemTemplate>
<asp:TextBox ID="txtCity" Width="100px" runat="server" Text='<%= Bind("VendorCity") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox ID="txtCity1" Width="100px" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Label3" runat="server" Text='<%= Bind("VendorCity") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="State" SortExpression="VendorState">
<EditItemTemplate>
<asp:TextBox ID="txtState" Width="100px" runat="server" Text='<%= Bind("VendorState") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox ID="txtState1" Width="100px" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Label4" runat="server" Text='<%= Bind("VendorState") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Country" SortExpression="VendorCountry">
<EditItemTemplate>
<asp:TextBox ID="txtCountry" Width="100px" runat="server" Text='<%= Bind("VendorCountry") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox ID="txtCountry1" Width="100px" runat="server" ></asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Label5" runat="server" Text='<%= Bind("VendorCountry") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="VendorDescription"
SortExpression="VendorDescription">
<EditItemTemplate>
<asp:TextBox ID="txtDescription" TextMode="MultiLine" runat="server" Text='<%= Bind("VendorDescription") %>'></asp:TextBox>
</EditItemTemplate>
<FooterTemplate>
<asp:TextBox ID="txtDescription1" runat="server" TextMode="MultiLine" >
</asp:TextBox>
</FooterTemplate>
<ItemTemplate>
<asp:Label ID="Label6" runat="server" Text='<%= Bind("VendorDescription") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>

<asp:TemplateField HeaderText="Edit" ShowHeader="False">
<EditItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="True"

```

```

CommandName="Update" Text="Update"></asp:LinkButton>
<asp:LinkButton ID="LinkButton2" runat="server" CausesValidation="False"
CommandName="Cancel" Text="Cancel"></asp:LinkButton>
</EditItemTemplate>
<FooterTemplate>
<asp:LinkButton ID="LinkButton2" runat="server" CausesValidation="False"
CommandName="AddNew" Text="Add New"></asp:LinkButton>
</FooterTemplate>
<ItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="False"
CommandName="Edit" Text="Edit"></asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
<asp:CommandField HeaderText="Delete" ShowDeleteButton="True" ShowHeader="True" />
<asp:CommandField HeaderText="Select" ShowSelectButton="True" ShowHeader="True" />

</Columns>
<RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
<FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
<PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
</asp:GridView>

```

Now build and run the application. The results looks like following.

VendorId	FirstName	LastName	City	State	Country	VendorDescription	Edit	Delete	Select
1	Raj	Beniwal	Boothwyn	PA	USA	this is vendor description one	Edit	Delete	Select
2	Nikita	Arora	Philadelphia	PA	USA	this is vendor description two	Edit	Delete	Select
3	Joe	Telmadge	Vargenia	VA	USA	this is vendor description three	Edit	Delete	Select
4	Alex	Trim	Alaska	AL	USA	this is vendor description four	Edit	Delete	Select
5	Den	Sem	newyork	NY	USA	deni is a american citizen	Edit	Delete	Select
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Add New		
1 2 3									

Figure 1.

Select command is use to select a particular row on select LinkButton click:

```

<asp:CommandField HeaderText="Select" ShowSelectButton="True" ShowHeader="True" />

```

Result looks like this on the select link.

VendorId	FirstName	LastName	City	State	Country	VendorDescription	Edit	Delete	Select
1	Raj	Beniwal	Boothwyn	PA	USA	this is vendor description one	Edit	Delete	Select
2	Nikita	Arora	Philadelphia	PA	USA	this is vendor description two	Edit	Delete	Select
3	Joe	Telmadge	Vargenia	VA	USA	this is vendor description three	Edit	Delete	Select
4	Alex	Trim	Alaska	AL	USA	this is vendor description four	Edit	Delete	Select
5	Den	Sem	newyork	NY	USA	deni is a american citizen	Edit	Delete	Select
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Add New		
1 2 3									

Figure 2.

This event is used for paging. As you can see from the code below, we simply set a new page index and rebind the data.

```
protected void ResultGridView_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    ResultGridView.PageIndex = e.NewPageIndex;
    FillVendorGrid();
}
```

This event shows how to delete a row on delete LinkButton click.

```
protected void ResultGridView_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    cmd.Connection = conn;
    cmd.CommandText = "DELETE FROM Vendor WHERE VendorId='" +
ResultGridView.DataKeys[e.RowIndex].Values[0].ToString() + "'";
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    FillVendorGrid();
}
```

This event is used to show a row in editable mode.

```
protected void ResultGridView_RowEditing(object sender, GridViewEditEventArgs e)
{
    ResultGridView.EditIndex = e.NewEditIndex;
    FillVendorGrid();
}
```

This event will update information in database.

```
protected void ResultGridView_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    TextBox txtFName = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtFName");
    TextBox txtLName = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtLName");
    TextBox txtCity = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtCity");
    TextBox txtState = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtState");
```



```

TextBox txtCountry = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtCountry");
TextBox txtDescription = (TextBox)ResultGridView.Rows[e.RowIndex].FindControl("txtDescription");

cmd.Connection = conn;
cmd.CommandText = "UPDATE Vendor SET VendorFName =" + txtFName.Text + ",VendorLName =" +
txtLName.Text + ",VendorCity =" + txtCity.Text + ",VendorState =" + txtState.Text + ",VendorCountry =" +
txtCountry.Text + ",VendorDescription =" + txtDescription.Text + " WHERE VendorId=" +
ResultGridView.DataKeys[e.RowIndex].Values[0].ToString() + """;
conn.Open();
cmd.ExecuteNonQuery();
ResultGridView.EditIndex = -1;
FillVendorGrid();
conn.Close();

}

```

Result will look like this :

VendorId	FirstName	LastName	City	State	Country	VendorDescription	Edit	Delete	Select
1	Raj	Beniwal	Boothwyn	PA	USA	this is vendor description one	Update Cancel		
2	Nikita	Arora	Philadelphia	PA	USA	this is vendor description two	Edit	Delete	Select
3	Joe	Telmadge	Vargenia	VA	USA	this is vendor description three	Edit	Delete	Select
4	Alex	Trim	Alaska	AL	USA	this is vendor description four	Edit	Delete	Select
5	Den	Sem	newyork	NY	USA	deni is a american citizen	Edit	Delete	Select
							Add New		

Figure 3.

This event is use to cancel editable model.

```

protected void ResultGridView_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    ResultGridView.EditIndex = -1;
    FillVendorGrid();
}

```

This RowCommand event is use to add a new row in database.

```

protected void ResultGridView_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName.Equals("AddNew"))
    {
        TextBox txtFName = (TextBox)ResultGridView.FooterRow.FindControl("txtFName1");
        TextBox txtLName = (TextBox)ResultGridView.FooterRow.FindControl("txtLName1");
        TextBox txtCity = (TextBox)ResultGridView.FooterRow.FindControl("txtCity1");
        TextBox txtState = (TextBox)ResultGridView.FooterRow.FindControl("txtState1");
        TextBox txtCountry = (TextBox)ResultGridView.FooterRow.FindControl("txtCountry1");
        TextBox txtDescription = (TextBox)ResultGridView.FooterRow.FindControl("txtDescription1");
        cmd.Connection = conn;
    }
}

```

```

cmd.CommandText = "INSERT INTO Vendor(VendorFName,
VendorLName, VendorCity, VendorState, VendorCountry, VendorDescription) Values('" + txtFName.Text + "', '" +
txtLName.Text + "', '" + txtCity.Text + "', '" + txtState.Text + "', '" + txtCountry.Text + "', '" + txtDescription.Text + "')";
conn.Open();
cmd.ExecuteNonQuery();
FillVendorGrid();
conn.Close();
}
}

```

For more information I am attaching my test application and database, so download it for better result and if you liked this article or you have any advice about my any article don't be hesitate drop me a comment on c-sharpcorner comment section.

Download 100% FREE Spire Office APIs

[ASP.NET 2.0](#) [ASP.NET2.0](#) [Edit Data](#) [Edit Data in GridView](#) [grid view](#) [gridview](#)
[GridView Control](#) [gridview in asp.net](#) [how to delete data using gridview](#)
[how to fetch data in GridView](#) [how to insert data using gridview](#)
[how to show data in GridView](#) [how to update data using gridview](#)
[how to use GridView in asp.net](#) [Show Data](#) [Show data in GridView](#)



RAJ KUMAR *TOP 50*

Raj Kumar is two time Microsoft MVP and four time C# Corner MVP. He is working as project manager with lots of hands on experience using ASP.NET, C#, MVC, AngularJS, Visual Basic .NET, SQL Server, WCF, HTML5, JQuery, Jav... [Read more](#)

<http://www.c-sharpcorner.com/members/raj-kumar-beniwal>

14
Rank

12.2m
Read

Platinum
Member

4
Times

2
Times

COMMENTS (40)

TRENDING UP

01 Using Generics With C#

02 Web Crawling With C#

What Is The Future Of XAML

03

04

[Best Practices For ASP.NET MVC Application](#)

05

[Best Programming Language Of 2016](#)

06

[Introduction To .NET Core 1.0](#)

07

[Factory Design Pattern Real World Example](#)

08

[Highest Paying Tech Jobs Of 2016](#)

09

[How To Become A Microsoft MVP](#)

10

[Keep Learning, Or Become Dust!](#)[View All](#)

Follow @csharpcorner

75.8K followers



C# Corner

520,526 likes

Like Page

Learn More

2 friends like this

[MVPs](#)[MOST VIEWED](#)[LEGENDS](#)[NOW](#)[PRIZES](#)[REVIEWS](#)[SURVEY](#)[DOWNLOADS](#)Hosted By [CBeyond Cloud Services](#)[ABOUT US](#)[FAQ](#)[MEDIA KIT](#)[MEMBERS](#)[STUDENTS](#)[LINKS](#)[CONTACT US](#)[PRIVACY POLICY](#)[TERMS & CONDITIONS](#)[SITEMAP](#)[REPORT A BUG](#)

©2016 C# Corner. All contents are copyright of their authors.