

تا قبل از ASP.NET 4.5 ، هنگام کار با GridView رسم بر این بوده که به خاصیت DataSource ، یک منبع داده (مانند SqlDataSource و ...) را Bind کرده و متد DataBind را صدا نموده و نتیجه نمایش داده می‌شد. اما با استفاده از ویژگی‌های جدید اضافه شده (هر چند با تأخیر نسبت به Grid های پیشرفته دیگر) کار با این کنترل راحت تر و خوانا تر شده است. یکی از این ویژگی‌ها را با هم بررسی می‌کنیم:

با استفاده از ویژگی SelectMethod میتوان متدی را به GridView معرفی کرد که وظیفه منبع داده را انجام داده و هنگام Bind فراخوانی شده و گرید را پر کند:

مثال:

```
<asp:GridView ID="gvCities"
    runat="server"
    AutoGenerateColumns="False"
    ItemType="WebApplication3.City"
    SelectMethod="GetAllCities">
    <Columns>
        <asp:TemplateField HeaderText="نام">
            <ItemTemplate><%#: Item.Name %></ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

نکته مهم در این کد ItemType است. با استفاده از این خاصیت به جای اینکه مانند قبل نام فیلدهایی که قرار است در گرید نمایش داده شود را بصورت string معرفی کنیم (مثلا در اینجا Eval("Name") ، اگر نام فیلد را غلط بنویسیم هنگام کامپایل خطایی صادر نمی‌شود)، آنرا بصورت Strongly Type نوشته و از اشتباه جلوگیری می‌کنیم. (+)

کد متد:

```
public IQueryable<City> GetAllCities()
{
    var context = new EFContext();
    var q = from c in context.City
            orderby c.Name
            select c;
    return q;
}
```

و سپس دستور زیر را فراخوانی می‌کنیم:

```
gvCities.DataBind();
```

اگر بخواهیم در گرید Paging داشته باشیم بصورت زیر عمل می‌کنیم:

```
<asp:GridView ID="gvCities"
    runat="server"
    AutoGenerateColumns="False"
    AllowPaging="True"
    PageSize="10"
    ItemType="WebApplication3.City"
    SelectMethod="GetAllCities">
    <Columns>
        <asp:TemplateField HeaderText="نام">
            <ItemTemplate><%#: Item.Name %></ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

که در اینجا دو خصوصیت AllowPaging و PageSize را مقدار دهی کرده ایم. این خصوصیت‌ها اجازه صفحه بندی را به گرید

می‌دهند. حال برای اینکه متد نیز برای صفحه بندی آماده شود باید سه آرگومان به آن اضافه کنیم: (نام پارامترها باید دقیقا موارد زیر باشد)

1- startRowIndex: نقطه شروع صفحه بندی را مشخص می‌کند.

2- maximumRows: تعداد سطرهایی که باید نمایش دهد را مشخص می‌کند.

3- totalRowCount: این پارامتر باید در تابع مقدار دهی شود (مانند مثال) تا مشخص شود نتیجه Query چند رکورد است و در نهایت باید تعداد صفحات را بر این اساس نمایش می‌دهد.

و برای اینکه صفحه بندی را در Query هم لحاظ کنیم از دو تابع Skip و Take استفاده شده است.

```
public IQueryable<City> GetAllCities(int startRowIndex, int maximumRows, out int totalRowCount)
{
    var context = new EFContext();
    var q = from c in context.City
            select c;

    totalRowCount = q.Count();

    return q.OrderBy(x=>x.Name).Skip(startRowIndex).Take(maximumRows);
}
```

نکته مهم در این متد IQueryable بودن آن است که باعث واکنشی داده‌ها بصورت صفحه به صفحه می‌شود. دستورات SQL تولید شده در پروفایلر:

Object context #3 [/WebForm1.aspx]

Statements Object context Usage

Short SQL

```
SELECT ... FROM (Select COUNT(1) AS [A1] From [dbo].[City] AS [...
SELECT ... FROM (Select [Extent1].[Id] AS [Id], [Extent1].[Name... WHERE [Extent1].[row_number] > 10
```

Details Stack Trace

```
1 SELECT TOP (10) [Extent1].[Id] AS [Id],
2 [Extent1].[Name] AS [Name]
3 FROM (SELECT [Extent1].[Id] AS [Id],
4 [Extent1].[Name] AS [Name],
5 row_number() OVER (ORDER BY [Extent1].[Name] ASC) AS [row_numbe
6 FROM [dbo].[City] AS [Extent1]) AS [Extent1]
7 WHERE [Extent1].[row_number] > 10
8 ORDER BY [Extent1].[Name] ASC
```

همانطور که مشاهده می‌کنید دو دستور SQL تولید شده ، یکی برای بازگرداندن تعداد رکوردها و یکی هم برای واکنشی داده‌ها به

اندازه تعداد رکوردهای مجاز در هر صفحه.

نظرات خوانندگان

نویسنده: saeid

تاریخ: ۱۳۹۱/۰۶/۳۰ ۱:۲۲

با سلام خدمت دوست عزیزم و تشکر از زحمت شما. سوالم این بود که واقعا این مورد در ورژن‌های قبلی امکان پذیر نبوده؟! <http://books.google.com/books?id=wAsB0IZFXV4C&pg=PA338&lpg=PA338&dq=SelectMethod+gridview+asp.net+startRowIndex+maximumRows+totalRowCount&source=bl&ots=sA1GqNuQr9&sig=vz0aA6EQ6h1w3i0TQ45dppyMwtw&hl=en&sa=X&ei=yztaUKTFJYSJhQf7o4CoDQ&ved=0CFAQ6AEwBQ#v=onepage&q=SelectMethod%20gridview%20asp.net%20startRowIndex%20maximumRows%20totalRowCount&f=false>

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۶/۳۰ ۸:۰۶

نه به صورت [Strongly Type Binding](#) که توضیح داده شد.

نویسنده: saeid

تاریخ: ۱۳۹۱/۰۶/۳۱ ۱۳:۲۴

بله درسته. ممنون از لینیکی که به اشتراک گذاشتید. راستش مطلب رو بطور کامل نخونده بودم.

همانطور که می‌دانید GridView جزء جداناپذیر از اکثر پروژه‌های برنامه نویسان ASP.NET Web forms می‌باشد. اکثراً روشی که در میان برنامه نویسان بیشتر استفاده می‌شود، قرار دادن یک دکمه/لینک در هر ردیف از GridView برای حذف رکورد مورد نظر می‌باشد. در این مقاله قصد دارم روشی را ارائه کنم تا کاربر قادر باشد هر تعداد رکورد را که مدنظر دارد، انتخاب کرده و با فشردن دکمه "حذف" رکوردهای انتخاب شده را حذف کند.

برای درک بهتر، ابتدا جدولی به اسم "Employee" را در SQL Server با مشخصات زیر ساخته :

```
CREATE TABLE [dbo].[Employee] (
    [EmpId] INT NOT NULL,
    [FirstName] VARCHAR (20) NOT NULL,
    [LastName] VARCHAR (20) NOT NULL,
    [City] VARCHAR (20) NOT NULL,
    PRIMARY KEY CLUSTERED ([EmpId] ASC)
);
```

1- یک GridView به صفحه افزوده و خاصیت AutoGenerateColumns آن را برابر False قرار دهید . 2- فیلدهایی را که قصد نمایش آنها در GridView را دارید به صورت زیر به GridView بیفزایید :

```
<asp:BoundField DataField="FirstName" HeaderText="First Name" />
```

3- برای قرار دادن کنترل‌های Asp.net که در اینجا منظور CheckBox می‌باشد می‌بایست از TemplateField و قرار دادن تگ ItemTemplate درون آن، به صورت زیر استفاده نمایید :

```
<asp:TemplateField>
    <ItemTemplate>
        <asp:CheckBox ID="chkDel" runat="server" />
    </ItemTemplate>
</asp:TemplateField>
```

و بعد از تگ GridView دکمه‌ای را برای حذف موارد انتخابی در فرم قرار دهید :

```
<asp:Button ID="btnDeleteRecord" runat="server" OnClick="btnDeleteRecord_Click" Text="Delete" />
```

برای نمایش یک پیغام به کاربر به منظور Confirm کردن دستور حذف در سمت کلاینت، قطعه کد Javascript زیر را قرار می‌دهیم:

```
function DeleteConfirm()
{
    var Ans = confirm("Do you want to Delete Selected Employee Record?");
    if (Ans)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

و در رویداد Page_Load کدهای زیر را جهت نمایش مقادیر در GridView و افزودن تابع فوق به دکمه، قبل از حذف رکوردها می‌افزاییم :

```
protected void Page_Load(object sender, EventArgs e)
```

```

{
    if(!IsPostBack)
    {
        //Displaying the Data
        showData();
        //Adding an Attribute to Server Control(i.e. btnDeleteRecord)
        btnDeleteRecord.Attributes.Add("onclick", "javascript:return DeleteConfirm()");
    }
}

```

```

//Method for Displaying Data
protected void showData()
{
    DataTable dt = new DataTable();
    SqlConnection con = new SqlConnection(cs);
    SqlDataAdapter adapt = new SqlDataAdapter("select * from Employee",con);
    con.Open();
    adapt.Fill(dt);
    con.Close();
    GridView1.DataSource = dt;
    GridView1.DataBind();
}

```

ابتدا تابع DeleteRecode را به صورت زیر پیاده سازی میکنیم :

که یک پارمتر را از ورودی دریافت میکند که ID رکورد انتخاب شده می باشد و با استفاده از ID، رکورد مورد نظر را حذف میکنیم :

```

protected void DeleteRecord(int empid)
{
    SqlConnection con = new SqlConnection(cs);
    SqlCommand com = new SqlCommand("delete from Employee where EmpId=@ID",con);
    com.Parameters.AddWithValue("@ID",empid);
    con.Open();
    com.ExecuteNonQuery();
    con.Close();
}

```

و اما بخش مهم مربوط به رویداد دکمه می باشد. در هنگام کلیک بر روی دکمه باید تمامی رکوردهای GridView را چک و تمامی رکوردهایی را که CheckBox آنها تیک خورده است گرفته و ID رکورد مورد نظر را به تابع DeleteRecode فرستاد و در پایان برای اعمال تغییرات، متد ShowDate را فراخوانی و GridView را مجدداً Bind می کنیم.

```

protected void btnDeleteRecord_Click(object sender, EventArgs e)
{
    foreach (GridViewRow grow in GridView1.Rows)
    {
        //Searching CheckBox("chkDel") in an individual row of Grid
        CheckBox chkdel = (CheckBox)grow.FindControl("chkDel");
        //If CheckBox is checked then delete the record with particular empid
        if(chkdel.Checked)
        {
            int empid = Convert.ToInt32(grow.Cells[1].Text);
            DeleteRecord(empid);
        }
    }
    //Displaying the Data in GridView
    showData();
}

```

نظرات خوانندگان

نویسنده: محمد حسین فخرآوری
تاریخ: ۱۸:۱۹ ۱۳۹۳/۰۷/۰۳

شرط بر اساس

where id in (.....)

در (chkdel.Checked)
شما ID بگیرید

نویسنده: عثمان رحیمی
تاریخ: ۲۱:۴۱ ۱۳۹۳/۰۷/۰۳

بله حق با شماست می‌توان کدهای فوق را نسبت به چیزی که گذاشته ام بهینه‌تر نوشت . ممنون از شما