

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:

The screenshot shows the 'Object context #3 [/WebForm1.aspx]' window with the 'Object context Usage' tab selected. It displays the 'Short SQL' and 'Details' of the generated query.

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:

```
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 لینکی که به اشتراک گذاشتید. راستش مطلب رو بطور کامل نخونده بودم.