

یکی از مسائلی که در هنگام کار با کنترل‌های داده‌ای نظیر GridView, ListView و .. با آن روبرو هستیم مسئله صفحه بندی می باشد و در بسیاری از موارد، کل اطلاعات در هر درخواست، بارگذاری میشود. در حالیکه روش بهینه به این صورت است که با توجه به PageSize و Index رکورد، می توان تعداد رکورد مورد نظر در همان صفحه را بارگذاری کرد، نه کل رکوردها را. در این مثال که از Ef Code First و الگوی Unit Of Work استفاده کرده ام، قصد نمایش اطلاعات در یک ListView و صفحه بندی آن را بصورت chunk chunk دارم. برای آشنایی بیشتر با الگوی Unit Of Work می توانید به مقاله [UOW](#) در همین سایت مراجعه کنید.

ابتدا یک کنترل ListView روی صفحه ایجاد می کنیم. برای آشنایی بیشتر با این کنترل و بررسی قابلیت های آن میتوان از این مقاله [معرفی کنترل های ListView و DataPager](#) استفاده کنید.

سپس با توجه به الگوی Unit Of Work از یک مدل ساده استفاده می کنیم. همچنین برای بارگذاری اطلاعات به صورت صفحه به صفحه، نیاز به داشتن Index رکورد و PageSize، جهت محاسبه تعداد رکورد مورد نیاز داریم.

```
public class User
{
    public Int64 UserId { get; set; }
    public String UserName { get; set; }
}
```

```
public interface IUserService
{
    int GetCustomerCount();
    List<User> GetCustomers(int StartIndex, int PageSize);
}
```

```
public class ImplUserService : IUserService
{
    IUnitOfWork _uow;
    IDbSet<User> _user;

    public ImplUserService(IUnitOfWork uow)
    {
        _uow = uow;
        _user = uow.Set<User>();
    }

    public int GetCustomerCount()
    {
        int totalCount = _user.ToList().Count;
        return totalCount;
    }

    public List<User> GetCustomers(int StartIndex, int PageSize)
    {
        return _user.OrderBy(i => i.UserId).Skip(StartIndex).Take(PageSize).ToList();
    }
}
```

در کدهای بالا متد GetCustomerCount تعداد کل رکوردهایی که باید واکشی شوند را مشخص می کند. همچنین می توان این تعداد را در cache یا ViewState ذخیره کرد تا در دفعات بعدی در صورت خالی نبودن مقدار Cache با ViewState، نیاز به محاسبه مجدد count نداشته باشیم.

متد GetCustomer که کار اصلی را انجام میدهد از دو پارامتر استفاده می کند: StartIndex نقطه شروع و PageSize تعداد رکورد مورد نظر. در اینجا از دستورات Linq استفاده شده و دستور Skip مشخص میکند از کدام شماره رکورد به بعد شروع به واکشی

نماید و دستور Take مشخص می‌کند که چه تعداد رکورد را واکشی نماید.

حالا به سراغ کدهای HTML می‌رویم. در آنجا علاوه بر ListView نیاز به DataPager جهت صفحه بندی و Object DataSource جهت کنترل بارگذاری اطلاعات به صورت chunk chunk داریم.

```
<asp:ListView ID="ListView1" runat="server" DataSourceID="ObjectDataSource1">
    <ItemTemplate>
        <tr style="background-color: #DCDCDC; color: #000000;">
            <td>
                <asp:Label ID="UserIdLabel" runat="server" Text='<%= Eval("UserId") %>' />
            </td>
            <td>
                <asp:Label ID="UserNameLabel" runat="server" Text='<%= Eval("UserName") %>' />
            </td>
        </tr>
    </ItemTemplate>

    <LayoutTemplate>
        <table runat="server">
            <tr runat="server">
                <td runat="server">
                    <table id="itemPlaceholderContainer" runat="server" border="1"
style="background-color: #FFFFFF;
border-collapse: collapse; border-color: #999999; border-style: none;
border-width: 1px;">
                        <tr runat="server" style="background-color: #DCDCDC; color: #000000;">
                            <th runat="server">
                                UserId
                            </th>
                            <th runat="server">
                                UserName
                            </th>
                        </tr>
                        <tr id="itemPlaceholder" runat="server">
                        </tr>
                    </table>
                </td>
            </tr>
            <tr runat="server">
                <td runat="server" style="text-align: center; background-color: #CCCCCC;
color: #000000;">
                    <asp:DataPager ID="DataPager1" runat="server" PageSize="2">
                        <Fields>
                            <asp:NextPreviousPagerField ButtonType="Button"
ShowFirstPageButton="True" ShowNextPageButton="False"
ShowPreviousPageButton="False" />
                            <asp:NumericPagerField />
                            <asp:NextPreviousPagerField ButtonType="Button"
ShowLastPageButton="True" ShowNextPageButton="False"
ShowPreviousPageButton="False" />
                        </Fields>
                    </asp:DataPager>
                </td>
            </tr>
        </table>
    </LayoutTemplate>
</asp:ListView>
```

همانطور که مشاهده میکنید در DataPager ، مقدار PageSize مشخص شده است. اما Object DataSource

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server" EnablePaging="True"
SelectCountMethod="GetCustomerCount"
SelectMethod="GetCustomers" TypeName="UserService.ImplUserService" EnableViewState="False"
MaximumRowsParameterName="PageSize" StartRowIndexParameterName="StartIndex">
```

که فکر می‌کنم با توجه به متدهای تعریف شده در بالا ، تعریف Object DataSource کاملاً گویا میباشد.

نکته مهم:

اگر الان برنامه را اجرا کنید با خطای No parameterless constructor defined for this object روبرو خواهید شد که جهت

حل این مشکل از کد زیر استفاده میکنیم.

```
protected void ObjectDataSource1_ObjectCreating(object sender, ObjectDataSourceEventArgs e)
{
    e.ObjectInstance = ObjectFactory.GetInstance<IUserService>();
}
```

در واقع نیاز است تا یک وهله از کلاس مورد نظر را به Object DataSource معرفی کنیم. حال با اجرای برنامه و Trace آن متوجه خواهید شد که با کلیک بر روی شماره صفحه، تنها به تعداد رکوردهای همان صفحه، واکشی خواهیم داشت.

با تشکر از راهنمایی‌های آقای نصیری، امیدوارم این مقاله برای دوستان مفید باشد. منتظر نظرات دوستان هستم و اینکه چه جوری بتوانیم اینکار رو با jquery ajax و به صورت سبکتر انجام بدیم.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۵ ۱۳۹۱/۰۴/۳۰

یک نکته تکمیلی: در اینجا بهتر است جهت محاسبه تعداد رکوردها از متد Count مستقیماً بر روی dbSet استفاده کنید (نیازی به ToList ندارد).

نویسنده: ایمان اسلامی
تاریخ: ۱۴:۳۴ ۱۳۹۱/۰۴/۳۰

بله کاملاً درسته
ممنون مهندس.

نویسنده: علیرضا اسم‌رام
تاریخ: ۱۵:۵۵ ۱۳۹۱/۰۴/۳۰

سلام و تشکر از شما بخاطر این مطلب. پیشنهاد می‌کنم جهت توسعه این ترفند (همانطور که اشاره کردید) نگاهی به [+](#) و [+](#) و [+](#) بیاندازید.
موفق باشید.

نویسنده: ایمان اسلامی
تاریخ: ۱۶:۳۰ ۱۳۹۱/۰۴/۳۰

با سلام و تشکر از شما بابت نظرتون
scrolling هم راه خوبیه برای اینکار که کاربردهای زیادی داره
ممنونم.