

## قسمت دوم

### 8. ORM Lazy Load

در هنگام استفاده از ORM ها دقت کنید کجا از [Lazy Load](#) استفاده می‌کنید. Lazy Load باعث می‌شود وقتی شما اطلاعات مرتبط را از بانک اطلاعات واکشی می‌کنید، این واکشی اطلاعات در چند query از بانک انجام شود. در عوض عدم استفاده از [Lazy Load](#) باعث می‌شود تمامی اطلاعات مورد نیاز شما در یک query از بانک اطلاعاتی دریافت شود. این موضوع یعنی سر بار کمتر در شبکه، در بانک اطلاعاتی، در منابع حافظه و منابع پر ارزش cpu در سرورها. البته استفاده از include در حالت فعال بودن یا نبودن lazy هم داستان مجزایی دارد که اگر عمری باقی باشد راجع به آن مقاله ای خواهیم نوشت.

به این نمونه دقت کنید:

```
List<Customer> customers = context.Customers.ToList();
foreach (Customer cust in context.Customers){
    Console.WriteLine("Customer {0}, Account {1}", cust.Person.LastName.Trim() + ", " +
        cust.Person.FirstName, cust.AccountNumber);
}
```

همچنین کدی (در صورت فعال بودن Lazy Load در ORM) در صورتی که جدول Customers دارای 1000 رکورد باشد، باعث می‌شود برنامه 1001 دستور sql تولید و در بانک اجرا گردد.

برای اطلاع بیشتر می‌توانید به این [مقاله](#) مراجعه نمایید.

### 9. استفاده از MiniProfiler

سعی کنید از MiniProfiler در تمامی پروژه‌ها استفاده کنید. البته وقتی نرم افزار را در اختیار مصرف کننده قرار می‌دهید، آن را غیر فعال کنید. می‌توانید از متغیرهای compiler برای مجزا کردن buildهای متفاوت در برنامه خود استفاده کنید:

```
#if DEBUG then
// فعال سازی MiniProfiler
#endif
```

ایده دیگری هم وجود دارد. شما می‌توانید MiniProfiler را برای کاربر Admin یا کاربر Debugger فعال و برای بقیه غیر فعال کنید. در باب MiniProfiler مسائلی زیادی وجود دارد که چند نمونه از آن در همین سایت در [این مقاله](#) و [این مقاله](#) در دسترس است. البته می‌توانید از ابزارهای دیگری مانند [Glimpse](#) که در این زمینه وجود دارد نیز استفاده کنید. لب کلام این نکته استفاده از profiler برای نرم افزار خود می‌باشد.

### 10. Data Paging در بانک اطلاعاتی

هنگامیکه از کامپوننت‌های شرکت‌های دیگر (Third party) استفاده می‌کنید، اطمینان حاصل کنید که صفحه بندی اطلاعات در بانک اطلاعاتی انجام می‌شود. برای نمونه کاپوننت گرید شرکت Telerik چند نوع صفحه بندی را پشتیبانی می‌کند. صفحه بندی سمت کاربر (توسط JavaScript)، صفحه بندی سمت سرور توسط کامپوننت و صفحه بندی مجازی. صفحه بندی سمت کاربر یعنی تمامی اطلاعات از سرور به کاربر فرستاده شده و در سمت کاربر عمل صفحه بندی انجام می‌شود. این یعنی واکشی تمامی اطلاعات از بانک و در مورد نرم افزارهای پرکاربر با حجم اطلاعات زیاد یعنی فاجعه. صفحه بندی سمت سرور ASP.NET هم یعنی واکشی اطلاعات از سرور بانک به سرور برنامه و سپس صفحه بندی توسط برنامه. این موضوع هم ممکن است مشکلات زیادی را ایجاد نماید چون باید حداقل تمامی رکوردها از اولین رکورد تا آخرین رکورد صفحه جاری از بانک واکشی شود که این عمل علاوه بر ایجاد سر بار شبکه، سر بار IO در بانک اطلاعاتی و سر بار cpu در سرور ASP.NET ایجاد می‌کند. استفاده از صفحه بندی مجازی، شما را قادر می‌کند بتوانیم اطلاعات را در بانک صفحه بندی کرده و فقط صفحه مورد نظر خود را از بانک واکشی کنیم.

این حالت مجازی در اکثر component ها که توسط شرکت های مختلف ایجاد شده وجود دارد ولی ممکن است نام های متفاوتی داشته باشد. برای این موضوع باید به راهنمای component خریداری شده مراجعه کنید و یا به فروم ها ... مراجعه نمایید.

#### 11. بررسی تعداد کوئری های صادر شده در یک صفحه و تعداد رکوردهای بازگشت داده شده توسط آنها

این [به این معنا نیست](#) که برای هر query یک context مجزا ایجاد کنید، منظور این است که به بهانه اینکه اطلاعات مختلفی از جداول مختلف مورد نیاز است، query خود را آن قدر پیچیده یا گسترده ننویسیم که یا process آن در بانک زمان و سر بار زیادی ایجاد کند و یا حجم اطلاعات بلا استفاده ای را از بانک به سرور برنامه لود نماید. به جای این موضوع می توانید در یک یا چند context دستورات مجزای واکنشی اطلاعات صادر کنید تا تنها اطلاعات مورد نیاز خود را واکنشی نمایید. البته این موضوع باعث نشود که تعداد query ها مثلا به 1000 عدد برسد! یعنی باید فیمابین query های پیچیده و query های ساده ولی با تعداد یکی را که مناسبتر با پروژه است انتخاب کنید که این موضوع با تجربه و تست حاصل می شود.

## نظرات خوانندگان

نویسنده: جواد

تاریخ: ۱۳۹۲/۰۵/۱۲ ۳:۱۵

"استفاده از صفحه بندی مجازی، شما را قادر می‌کند بتوانیم اطلاعات را در بانک صفحه بندی کرده و فقط صفحه مورد نظر خود را از بانک واکنشی کنیم. این حالت مجازی در اکثر componentها که توسط شرکت‌های مختلف ایجاد شده وجود دارد"

میشه این رو بیشتر توضیح بدید که منظورتون چیه. یا باید تمامی اطلاعات رو بفرستیم بعد صفحه بندی کنه یا اینکه به ازای هر صفحه یک کوئری به بانک بفرسته و اطلاعات رو نشون بده. حالا این صفحه بندی مجازی کجا کاربرد داره.

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۵/۱۲ ۱۵:۱۷

هیچ کامپوننتی وجود خارجی نداره که قسمت مدیریت سمت بانک اطلاعاتی رو هم خودش به تنهایی انجام بده. همین کنترل‌های پیش فرض ASP.NET رو هم اگر ازشون درست استفاده کنیم، مشکلات کارایی ندارند. مثلاً: (نکته مهمش Skip.Take.ToList استفاده شده هست)

### واکنشی اطلاعات به صورت chunk chunk (تکه تکه) و نمایش در ListView

نویسنده: مرتضی

تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۱:۲۴

چرا وجود نداره!  
از گرید Kendo استفاده کنید اگر paging رو فعال کنید خودش مدیریت میکنه

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۲:۲۷

بله. مدیریت می‌کنه، نه به تنهایی. اینجا هم باید اطلاعات Skip و Take رو بهش بدی تا صفحه بندی کم هزینه‌ای رو اعمال کنه. خود GridView در وب فرم‌ها هم paging داره. مشکلیش اینه که در حالت پیش فرض کل اطلاعات رو از سرور واکنشی می‌کنه و بعد یک صفحه رو نمایش می‌ده. بحث اینه که گرید اگر قراره یک صفحه رو نمایش بده که 20 ردیف داره، بتونه فقط 20 رکورد رو واکنشی کنه و نه کل اطلاعات رو و این نیاز به کوئری‌های خاصی در سمت سرور داره. یک نمونه‌اش رو در واکنشی اطلاعات به صورت تکه تکه لینک دادم.

نویسنده: م منفرد

تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۲:۴۶

1. اکثر کنترل‌های ASP.NET WebForm قابلیت bind به DataSet را دارد. اگر از آنها در این مدل استفاده نمایید کار صفحه بندی به عهده کنترل+DataSet می‌افتد.
  2. صفحه بندی مجازی یعنی شما به کنترل می‌گویید کل اطلاعات شما مثلاً 51 صفحه است، الان صفحه 4 را نمایش می‌دهی این هم اطلاعات صفحه 4. بعد وقتی کاربر بر روی گزینه صفحه بعد کلیک کرد، به کنترل می‌گویید کل اطلاعات 51 صفحه است، الان صفحه 5 را نمایش می‌دهی و این هم اطلاعات آن.
- برای مثال می‌توانی به این مثال در DataTables مراجعه کنید

نویسنده: مرتضی

تاریخ: ۱۳۹۲/۰۵/۱۳ ۰:۱۳

گفتم که خودش مدیریت می‌کنه یعنی اینکه اطلاعات Skip , Take رو نمی‌خواد بهش بدی و خودش اینکار رو انجام میده - من دارم ازش تو پروژم استفاده میکنم - کل اطلاعات رو واکنشی نمی‌کنه و همون 20 رکورد فرضا صفحه 3 رو واکنشی میکنه

نویسنده: محسن خان  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۰:۳۱

می‌تونم یک مثال با پروفایل SQL نهایی آن ارائه بدی. مطابق بررسی که کردم و حداقل دو تا لینکی که دادم در مورد این کتابخانه، موارد پردازش Skip و Take سمت سرور اون خودکار نیست.

نویسنده: مرتضی  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۴:۴۹

کد Razor - کد VB - خروجی SQL

```
@(Html.Kendo.Grid(Of Models.vProject).Name("ProjectsGrid") _
    .Columns(Sub(column)
        With column
            .Bound(Function(p) p.ProjectId).Hidden()
            .Bound(Function(p) p.Supervisor).Title("ناظر")
            .Bound(Function(p) p.MapNumber).Title("شماره نقشه")
            .Bound(Function(p) p.MapCode).Title("کد نقشه")
            .Bound(Function(p) p.NewStructureArea).Title("متراژ")
            .Bound(Function(p) p.NumberOfFloors).Title("تعداد طبقات")
            .Bound(Function(p) p.InsuranceName).Title("بیمه")
        End With
    End Sub).Pageable(Sub(p)
        p.Enabled(True)
        p.Info(True)
        p.PageSizes(True)
        p.Messages(Sub(m)
            m.Display("{0} - {1} رکورد {2} از")
            m.Empty("رکوردی برای نمایش وجود ندارد")
            m.Of("از")
            m.Page("صفحه")
            m.ItemsPerPage("رکورد در هر صفحه")
            m.Refresh("بروزرسانی")
        End Sub)
    End Sub).Selectable(Sub(s)
        s.Enabled(True).Mode(GridSelectionMode.Single).Type(GridSelectionType.Row)) _
        .DataSource(Sub(ds)
            ds.Ajax.ServerOperation(True).Read("GetProjects", "Home") _
                .Model(Sub(m) m.Id(Function(modelId)
                    modelId.ProjectId)))
        )
```

```
<HttpPost>
Function GetProjects(<DataSourceRequest> request As DataSourceRequest) As JsonResult
    Return Json(db.vProjects.ToDataSourceResult(request))
End Function
```

```
SELECT TOP (5) [Extent1].[ProjectId] AS [ProjectId],
               [Extent1].[Supervisor] AS [Supervisor],
               [Extent1].[MapCode] AS [MapCode],
               [Extent1].[MapNumber] AS [MapNumber],
               [Extent1].[EmployerName] AS [EmployerName],
               [Extent1].[InsuranceName] AS [InsuranceName],
               [Extent1].[NewStructureArea] AS [NewStructureArea],
               [Extent1].[NumberOfFloors] AS [NumberOfFloors]
FROM (SELECT [Extent1].[ProjectId] AS [ProjectId],
             [Extent1].[Supervisor] AS [Supervisor],
             [Extent1].[MapCode] AS [MapCode],
             [Extent1].[MapNumber] AS [MapNumber],
             [Extent1].[EmployerName] AS [EmployerName],
             [Extent1].[InsuranceName] AS [InsuranceName],
             [Extent1].[NewStructureArea] AS [NewStructureArea],
             [Extent1].[NumberOfFloors] AS [NumberOfFloors],
             row_number() OVER (ORDER BY [Extent1].[ProjectId] ASC) AS [row_number]
      FROM (SELECT [vProject].[ProjectId] AS [ProjectId],
                  [vProject].[Supervisor] AS [Supervisor],
```

```
[vProject].[MapCode] AS [MapCode],
[vProject].[MapNumber] AS [MapNumber],
[vProject].[EmployerName] AS [EmployerName],
[vProject].[InsuranceName] AS [InsuranceName],
[vProject].[NewStructureArea] AS [NewStructureArea],
[vProject].[NumberOfFloors] AS [NumberOfFloors]
FROM [dbo].[vProject] AS [vProject]) AS [Extent1]) AS [Extent1]
WHERE [Extent1].[row_number] > 5
ORDER BY [Extent1].[ProjectId] ASC
```

نویسنده: محسن خان  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۸:۲۰

کد سمت سرور db.vProjects.ToDataSourceResult چطور تهیه شده؟ در موردش [اینجا بحث شده](#) . شما یک IQueryable باید در اختیارش قرار بدی (که از لحاظ لایه بندی کار مشکل داره) تا بر اساس اطلاعات شماره صفحه و غیره‌ای که از کلاینت می‌رسه خودش مباحث Take و Skip رو پیاده سازی کنه. در حقیقت این کتابخانه فقط [یک متد الحاقی](#) اضافه‌تر برای اینکار جهت مدیریت مباحث سمت سرور داره.

نویسنده: مرتضی  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۳:۴۵

درسته محسن خان- متد الحاقی ToDataSourceResult در خواست رو می‌گیره و.... بحث سر این بود که هیچ کامپوننتی وجود خارجی نداره که قسمت مدیریت سمت بانک اطلاعاتی رو هم خودش به تنهایی انجام بده

آره شیء DataSourceRequest شامل PageNumber , PageSize , .... می‌شه ولی دیگه ما کاری بهش نداریم و خودش صفحه بندی رو انجام می‌ده حالا به طریقی نمیشود گفت اگر ما از IQueryable استفاده کردیم حتما لایه بندی ما مشکل داره

نویسنده: محسن خان  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۴:۱۷

داره. [بهش می‌گن leaky abstraction](#) .

نویسنده: مرتضی  
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۴:۲۴

درسته بهش می‌گن [leaky abstraction](#) اما نمی‌گن 100% ایراد