

قسمتی از یک پروژه به همراه کلاس SqlHelper آن در کامنت‌های مطلب «[اهمیت Code review](#)» توسط یکی از خوانندگان بلاگ جهت Code review مطرح شده که بهتر است در یک مطلب جدید و مجزا به آن پرداخته شود. قسمت مهم آن کلاس SqlHelper است و مابقی در اینجا ندید گرفته می‌شوند:

```
//It's only for code review purpose!
using System.Data;
using System.Data.SqlClient;
using System.Web.Configuration;

public sealed class SqlHelper
{
    private SqlHelper() { }

    // Send Connection String
    //-----
    public static string GetCntString()
    {
        return WebConfigurationManager.ConnectionStrings["db_ConnectionString"].ConnectionString;
    }

    // Connect to Data Base SqlServer
    //-----
    public static SqlConnection Connect2Db(ref SqlConnection sqlCnt, string cntString)
    {
        try
        {
            if (sqlCnt == null) sqlCnt = new SqlConnection();
            sqlCnt.ConnectionString = cntString;
            if (sqlCnt.State != ConnectionState.Open) sqlCnt.Open();
            return sqlCnt;
        }
        catch (SqlException)
        {
            return null;
        }
    }

    // Run ExecuteScalar Command
    //-----
    public static string RunExecuteScalarCmd(ref SqlConnection sqlCnt, string strCmd, bool blnClose)
    {
        Connect2Db(ref sqlCnt, GetCntString());
        using (sqlCnt)
        {
            using (SqlCommand sqlCmd = sqlCnt.CreateCommand())
            {
                sqlCmd.CommandText = strCmd;
                object objResult = sqlCmd.ExecuteScalar();
                if (blnClose) CloseCnt(ref sqlCnt, true);
                return (objResult == null) ? string.Empty : objResult.ToString();
            }
        }
    }

    // Close SqlServer Connection
    //-----
    public static bool CloseCnt(ref SqlConnection sqlCnt, bool nullSqlCnt)
    {
        try
        {
            if (sqlCnt == null) return true;
            if (sqlCnt.State == ConnectionState.Open)
```

```

        {
            sqlCnt.Close();
            sqlCnt.Dispose();
        }
        if (nullSqlCnt) sqlCnt = null;
        return true;
    }
    catch (SqlException)
    {
        return false;
    }
}
}

```

مثالی از نحوه استفاده ارائه شده:

```

protected void BtnTest_Click(object sender, EventArgs e)
{
    SqlConnection sqlCnt = new SqlConnection();
    string strQuery = "SELECT COUNT(UnitPrice) AS PriceCount FROM [Order Details]";

    // در این مرحله پارامتر سوم یعنی کانکشن باز نگه داشته شود
    string strResult = SqlHelper.RunExecuteScalarCmd(ref sqlCnt, strQuery, false);

    strQuery = "SELECT LastName + N'-' + FirstName AS FullName FROM Employees WHERE (EmployeeID
= 9)";
    // در این مرحله پارامتر سوم یعنی کانکشن بسته شود
    strResult = SqlHelper.RunExecuteScalarCmd(ref sqlCnt, strQuery, true);
}

```

مروری بر این کد:

## 1) نحوه کامنت نوشتن

بین سی شارپ و زبان سی++ تفاوت وجود دارد. این نحوه کامنت نویسی بیشتر در سی++ متداول است. اگر از ویژوال استودیو استفاده می‌کنید، مکان نما را به سطر قبل از یک متد منتقل کرده و سه بار پشت سر هم forward slash را تایپ کنید. به صورت خودکار ساختار خالی زیر تشکیل خواهد شد:

```

/// <summary>
///
/// </summary>
/// <param name="sqlCnt"></param>
/// <param name="cntString"></param>
/// <returns></returns>
public static SqlConnection Connect2Db(ref SqlConnection sqlCnt, string cntString)

```

این روش مرسوم کامنت نویسی کدهای سی شارپ است. خصوصاً اینکه ابزارهایی وجود دارند که به صورت خودکار از این نوع کامنت‌ها، فایل CHM درست می‌کنند.

## 2) وجود سازنده private

احتمالاً هدف این بوده که نه شخصی و نه حتی کامپایلر، وهله‌ای از این کلاس را ایجاد نکند. بنابراین بهتر است کلاسی را که تمام متدهای آن static است (که به این هم خواهیم رسید!)، راساً static معرفی کنید. به این ترتیب نیازی به سازنده private

نخواهد بود.

### (3) وجود try/catch

یک اصل کلی وجود دارد: اگر در حال طراحی یک کتابخانه پایه‌ای هستید، try/catch را در هیچ متدی از آن لحاظ نکنید. بله؛ درست خوندید! لطفاً try/catch ننویسید! کرش کردن برنامه خوب است! لایه‌های بالاتر برنامه که در حال استفاده از کدهای شما هستند متوجه خواهند شد که مشکلی رخ داده و این مشکل توسط کتابخانه مورد استفاده «خفه» نشده. برای مثال اگر هم اکنون SQL Server در دسترس نیست، لایه‌های بالاتر برنامه باید این مشکل را متوجه شوند. Exception اصلاً چیز بدی نیست! کرش برنامه اصلاً بد نیست! فرض کنید که دچار بیماری شده‌اید. اگر مثلاً تبی رخ ندهد، از کجا باید متوجه شد که نیاز به مراقبت پزشکی وجود دارد؟ اگر هیچ علامتی بروز داده نشود که تا الان نسل بشر منقرض شده بود!

### (4) وجود ref و out

دوستان گرامی! این ref و out فقط جهت سازگاری با زبان C در سی شارپ وجود دارد. لطفاً تا حد ممکن از آن استفاده نکنید! مثلاً استفاده از توابع API ویندوز که با C نوشته شده‌اند. یکی از مهم‌ترین کاربردهای pointers در زبان سی، دریافت بیش از یک خروجی از یک تابع است. برای مثال یک متد API ویندوز را فراخوانی می‌کنید؛ خروجی آن یک ساختار است که به کمک pointers به عنوان یکی از پارامترهای همان متد معرفی شده. این روش به وفور در طراحی ویندوز بکار رفته. ولی خوب در سی شارپ که از این نوع مشکلات وجود ندارد. یک کلاس ساده را طراحی کنید که چندین خاصیت دارد. هر کدام از این خاصیت‌ها می‌توانند نمایانگر یک خروجی باشند. خروجی متد را از نوع این کلاس تعریف کنید. یا برای مثال در دات نت 4، امکان دیگری به نام Tuples معرفی شده برای کسانی که سریع می‌خواهند چند خروجی از یک تابع دریافت کنند و نمی‌خواهند برای اینکار یک کلاس بنویسند. ضمن اینکه برای مثال در متد Connect2Db، هم کانکشن یکبار به صورت ref معرفی شده و یکبار به صورت خروجی متد. اصلاً نیازی به استفاده از ref در اینجا نبوده. حتی نیازی به خروجی کانکشن هم در این متد وجود نداشته. کلیه تغییرات شما در شیء کانکشنی که به عنوان پارامتر ارسال شده، در خارج از آن متد هم منعکس می‌شود (شبيه به همان بحث pointers در زبان سی). بنابراین وجود ref غیرضروری است؛ وجود خروجی متد هم به همین صورت.

### (5) استفاده از using در متد RunExecuteScalarCmd

استفاده از using خیلی خوب است؛ همیشه اینکار را انجام دهید! اما اگر اینکار را انجام دادید، بدانید که شیء sqlCnt در پایان بدنه using، توسط GC نابوده شده است. بنابراین اینجا bool bInClose دیگر چه کاربردی دارد؟! تصمیم شما دیگر اهمیتی نخواهد داشت؛ چون کار تخریبی پیشتر انجام شده.

### (6) متد CloseCnt

این متد زاید است؛ به دلیلی که در قسمت (5) عنوان شد. using های استفاده شده، کار را تمام کرده‌اند. بنابراین بستن اشیاء dispose شده معنا نخواهد داشت.

### (7) در مورد نحوه استفاده

اگر SqlHelper را در اینجا مثلاً یک DAL ساده فرض کنیم (data access layer)، جای قسمت BLL (business logic layer) در اینجا خالی است. عموماً هم چون توضیحات این موارد را خیلی بد ارائه داده‌اند، افراد از شنیدن اسم آن‌ها هم وحشت می‌کنند. BLL یعنی کمی دست به Refactoring بزنید و این پیاده سازی منطق تجاری ارائه شده در متد BtnTest\_Click را به یک کلاس مجزا خارج از code behind پروژه منتقل کنید. Code behind فقط محل استفاده نهایی از آن باشد. همین! فعلاً با همین مختصر شروع کنید.

مورد دیگری که در اینجا باز هم مشهود است، عدم استفاده از پارامتر در کوئری‌ها است. چون از پارامتر استفاده نکرده‌اید، SQL Server مجبور است برای حالت EmployeeID = 9 یکبار execution plan را محاسبه کند، برای کوئری بعدی مثلاً EmployeeID = 19، اینکار را تکرار کند و الی آخر. این یعنی مصرف حافظه بالا و همچنین سرعت پایین انجام کوئری‌ها. بنابراین اینقدر در قید و بند باز نگه داشتن یک کانکشن نباشید؛ مشکل اصلی جای دیگری است!

### (8) برنامه وب و اطلاعات استاتیک!

این پروژه، یک پروژه ASP.NET است. دیدن تعاریف استاتیک در این نوع پروژه‌ها یک علامت خطر است! در این مورد قبلاً مطلب نوشتیم:

[متغیرهای استاتیک و برنامه‌های ASP.NET](#)

### یک درخواست عمومی!

لطف کنید در پروژه‌های «جدید» خودتون این نوع کلاس‌های SqlHelper رو «دور بریزید». یاد گرفتن کار با یک ORM جدید اصلاً سخت نیست. مثلاً طراحی Entity framework میکروسافت به حدی ساده است که هر شخصی با داشتن بهره‌هوشی در حد یک عنکبوت آبی یا حتی جلبک دریایی هم می‌تونه با اون کار کنه! فقط NHibernate هست که کمی مرد افکن است و گرنه مابقی به عمد ساده طراحی شده‌اند.

مزایای کار کردن با ORM ها این است:

- کوئری‌های حاصل از آن‌ها «پارامتری» است؛ که این دو مزیت عمده را به همراه دارد:

امنیت: مقاومت در برابر SQL Injection

سرعت و همچنین مصرف حافظه کمتر: با کوئری‌های پارامتری در SQL Server همانند رویه‌های ذخیره شده رفتار می‌شود.

- عدم نیاز به نوشتن DAL شخصی پر از باگ. چون ORM یعنی همان DAL که توسط یک سری حرفه‌ای طراحی شده.

- یک دست شدن کدها در یک تیم. چون همه بر اساس یک اینترفیس مشخص کار خواهند کرد.

- امکان استفاده از امکانات جدید زبان‌های دات نتی مانند LINQ و نوشتن کوئری‌های strongly typed تحت کنترل کامپایلر.

- پایین آوردن هزینه‌های آموزشی افراد در یک تیم. مثلاً EF را می‌شود به عنوان یک پیشنهاد در نظر گرفت؛ عمومی است و همه گیر. کسی هم از شنیدن نام آن تعجب نخواهد کرد. کتاب(های) آموزشی هم در مورد آن زیاد هست.

و ...

## نظرات خوانندگان

نویسنده: M\_royasaz  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۷:۱۹

"برنامه وب و اطلاعات استاتیک!" این مسئله شامل متدهای استاتیک در لایه بیزینس هم میشه ؟

نویسنده: hossein moradinia  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۸:۲۱

سلام

گفته شد که از out استفاده نکنیم. پس تکلیف استفاده از دستوری TryParse در سی شارپ چه میشود؟!  
جایگزینی برای آن وجود دارد یا اینکه باید استفاده شود?!

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۸:۲۵

طراحی TryParse مربوط به دات نت یک است که هنوز حال و هوای دوران C برقرار بود. اگر امروز میخواستند آن را طراحی کنند هیچ وقت از Out در آن استفاده نمی شد.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۸:۲۸

اگر فقط متدها استاتیک باشد، خیر. مانند مثال بالا. اما کیفیت این کد طوری است که تمایل به استفاده از اطلاعات استاتیک در آن بالا است. احتمالاً شاید چون شیک تر به نظر می رسد. در اون صورت اگر جایی نوشته شده `public static bool IsAdmin` یعنی تمام کاربران سایت هم اکنون ادمین هستند یا می توانند باشند.

نویسنده: Shima 2012  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۸:۳۳

WOW Wonderful

با سلام و تشکر از شما استاد گرامی بابت وقت گرانبهائی که برای آموزش افرادی مانند من هزینه میکنید. در ابتدا عرض کنم ارزش لطف شما را به خوبی میدانم چرا که برنامه نویس بزرگی مثل شما به جای وقت گذاشتن برای کدهای امثال من میتواند به Business خود پرداخته و در همین زمان صرف شده... (اجرکم عندالله)

خواهش دیگری که دارم این است که میتوانید یک سمپل از یکی از کارهایی که از لحاظ فنی مورد تأیید شما میباشد را جهت دانلود معرفی کنید تا من و امثال من بتوانیم از آن جهت یادگیری استفاده نماییم؟ (اگر از کارهای خودتان باشد که دیگر ...)

در هر صورت ممنونم از لطف شما.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۸:۴۳

اول نیاز به پایه تئوری هست برای کار عملی. از اینجا (^) می شود شروع کرد.

نویسنده: K Khodaei  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۹:۳۶

مشکل سرعت ORM را چگونه برطرف کنیم؟؟ یکی از مزایای خود datareader در وب سرعت آن هستش. با دور انداختن SqlHelper می توان این درخواست شما را عملی کرد؟؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۰۹:۵۹:۰۰

این سربرابر اینقدر نیست که اهمیتی داشته باشد. فقط قرار است یک کوئری LINQ به معادل SQL آن ترجمه شود. خیلی سریع است. همچنین امکان تهیه Compiled linq queries هم وجود دارد ([^](#)).  
ضمن اینکه مثلا NHibernate قابلیتی دارد به نام second level cache که اساسا برای پروژه‌های وب طراحی شده. قابلیت کش در سطح کوئری یا اطلاعات پرکاربرد و عمومی سایت را به صورت خودکار دارد. در موردش قبلا مطلب نوشتم: ([^](#)). سطح اول کش آن هم پیاده سازی حرفه‌ای همین باز نگه داشتن کانکشنی است که در کد SqlHelper بالا نویسنده موفق به پیاده سازی آن نشده، به علاوه کاهش رفت و آمدها به سرور: ([^](#))  
به علاوه NHibernate یک قابلیت دیگر هم دارد به نام ToFuture که می‌تونه چندین کوئری رو در طی یک رفت و برگشت برای شما انجام بده ([^](#)).  
و ... خیلی از best practices دیگر هم در آن لحاظ شده. خلاصه اینکه توانایی‌های بسیار ارزنده‌ای رو با عدم استفاده از ORMs از دست خواهید داد. منجمله همان بحث کوئری‌های پارامتری که عموما از نوشتن آن طفره می‌روند اما اینجا به صورت خودکار برای شما انجام می‌شود.

نویسنده: محمد صاحب  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۰:۱۱:۱۰

خیلی ممنون عالی بود.  
در رابطه با بحث Exception تو کلاس های پایه اگه Exception مجددا Throw بشه (با اضافه شدن کمی اطلاعات) این کار تو این کلاس پایه قابل توجهیه؟

یه همچین بحثی اینجا شده.  
<http://social.msdn.microsoft.com/Forums/en-US/vbgeneral/thread/5092363e-649d-45a6-8ae1-e9cf9f6db867>

یه سوال بی ربط: با چه روشی شما به مطالب تو کامنت ها لینک میدید من کامنتم رو از Word هم میارم لینک به مطالب حذف میشه!

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۰:۳۰:۱۹

- اگر مجددا throw بشه مشکلی نداره. مثلا اگر به کدهای کتابخانه NHibernate استفاده کنید، گاهی از اوقات از این روش استفاده کرده، مثلا برای ارائه پیغام واضح‌تری به کاربر اگر پیغام exception اولیه مفهوم نیست. یا می‌خواهید یک Exception کلی را لاگ کنید اما یک نمونه ساده‌تر را مثلا به دلایل امنیتی به کاربر نمایش دهید. اما حتما این لاگ کردن اولیه باید لحاظ شود چون فوق العاده در طول زمان کیفیت کد را بالا می‌برد و مشکلات را نمایان می‌کند.  
در کل یک کلاس پایه تا حد امکان نباید try/catch داشته باشد. سطح‌های بالاتر باید در این مورد تصمیم گیری کنند.  
- من تگ a href را آخری دستی درست می‌کنم.

نویسنده: Samin\_c2700  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۵:۲۷:۲۲

سلام  
از مطالب مفید و پربارتون ممنون .  
در مورد استفاده از EF یه مشکلی که برای من وجود داره و نتونستم براش یه راه حل درست و درمون پیدا کنم نحوه‌ی استفاده از EF با سایر بانک‌های اطلاعاتی (مثلا اراکله) که خود شما هم تو نقدی که بر کتاب آقای راد نوشته بودید ذکر کردیدش. من هرچی تو نت گشتم تنها راه حلی رو که برای اینکار گفته بودند استفاده از پروایدر های شرکت های ثالث بوده. آیا برای اینکار یه راه حل استاندارد وجود داره ؟

نویسنده: Vb\_asp\_net  
تاریخ: ۱۷:۳۹:۰۲ ۱۳۹۰/۱۰/۲۱

استاد من در یک وب سایت از متد های استاتیک استفاده کردم ولی زمانی که تعداد کاربرام خیلی زیاد شدند تداخل اطلاعات به وجود اومد ولی زمانی که اونا رو از حالت استاتیک خارج کردم دیگه همچین مشکلی به وجود نیامد.

نویسنده: وحید نصیری  
تاریخ: ۱۷:۴۵:۴۵ ۱۳۹۰/۱۰/۲۱

باید کدهای شما رو ببینم. اگر این بین اطلاعات اشتراکی استاتیک وجود داشته حتما مشکل را بوده.

نویسنده: Vb\_asp\_net  
تاریخ: ۱۷:۵۴:۱۴ ۱۳۹۰/۱۰/۲۱

```

کد من با زبان VB.Net هست که زیاد تفاوتی نداره من کد خودم رو اینجا می زارم
(Public Shared Function GetPAGES() As List(Of EntityPAGES
(Dim cn As New SqlConnection(SiteHelper.GetConnectionString
{Dim cmd As New SqlCommand("GET_PAGES", cn) With {.CommandType = CommandType.StoredProcedure
    ("", "")cmd.Parameters.AddWithValue'
    ()(Dim retlist As New List(Of EntityPAGES
        Dim reader As SqlDataReader = Nothing
        Try
            ()cn.Open
        ()reader = cmd.ExecuteReader
        If reader.HasRows Then
            Dim row As Integer = 1
            ()Do While reader.Read
                ()Dim item As New EntityPAGES
                (()item.Division.Division_id = Integer.Parse(reader("Division_id").ToString
                ()item.Division.Name_persian = reader("DIVISION_NAME").ToString
                (()item.Page_id = Integer.Parse(reader("Page_id").ToString
                (()item.Page_no = Integer.Parse(reader("Page_no").ToString
                (()item.Masterpage.Masterpage_id = Integer.Parse(reader("Masterpage_id").ToString
                ()item.Page_file_name = reader("Page_file_name").ToString
                ()item.Page_title = reader("Page_title").ToString
                ()item.Page_link = reader("Page_link").ToString
                (()item.Page_delete = Boolean.Parse(reader("Page_delete").ToString
                (()item.Active = Boolean.Parse(reader("Active").ToString
                ()item.Remark = reader("Remark").ToString
                (retlist.Add(item
            Loop
        End If
    Catch e1 As SqlException
        Throw
    Catch e2 As Exception
        Throw
    Finally
        If reader IsNot Nothing Then

```

```

(reader.Close
End If
If cn.State <> ConnectionState.Closed Then
(cn.Close
(cmd.Dispose
End If
End Try
Return retlist
End Function

```

استاد مثلاً این کدی که من نوشتم اگر تعداد زیادی کاربر در حال DataEntry باشند اطلاعات اونها با هم قاطی میشه.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۸:۰۰:۳۸

مایکروسافت پشتیبانی از پروایدر ADO.NET مرتبط به اوراکل را چندسالی است که رسماً قطع کرده و خود اوراکل این پروایدر رو داره ارائه می‌ده. در مورد EF هم به همین صورت: (^) و (^).

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۸:۲۴:۴۴

نه. این متد استاتیک مشکلی نداره و مشکل از اینجا نیست. بهتر است تراکنش‌ها رو لحاظ کنید تا اطلاعات تداخل نکنند. همچنین من اینجا در رویه ذخیره شده GET\_PAGES پارامتری یا آرگومانی نمی‌بینم که مقدار دهی شده باشه. در کل برای بررسی آن نیاز به مشاهده اطلاعات بیشتری هست مثلاً GET\_PAGES چی هست یا اینکه این متد بالا کجا فراخوانی میشه. آیا بلافاصله بعد از ورود اطلاعات هست؟ اگر اینطور است هم تراکنش نیاز دارد و هم GET\_PAGES نیاز به یک آرگومان یا پارامتر ورودی که مشخص کند، چه مواردی را باید فیلتر کند و نمایش دهد.

نویسنده: Shima2012  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۹:۱۳:۱۳

ممنون استاد در این زمینه کتابهای خوب فارسی هم سراغ دارید؟  
همچنین برای یادگیری MVC چه کتاب فارسی رو پیشنهاد میدید؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۱ ۱۹:۲۸:۰۸

یک سری ویدیوی آموزشی به روز و با کیفیت در این زمینه از سایت pluralsight موجود است. در گوگل جستجو کنید قابل دریافت هستند.

نویسنده: Tahery Naser  
تاریخ: ۱۳۹۰/۱۰/۲۲ ۰۰:۱۲:۱۰

مثلاً طراحی Entity framework مایکروسافت به حدی ساده است که هر شخصی با داشتن بهره هوشی در حد یک عنکبوت آبی یا حتی جلبک دریایی هم می‌تونه با اون کار کنه! این جمله کاملاً تایید میشه چون من یادش گرفتم و خیلی به این ORM ها علاقه مند شدم کار انجام میدن در حد معجزه.  
آقای نصیری آیا منبعی فارسی برای NHibernate وجود داره؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۰/۱۰/۲۲ ۰۰:۱۹:۵۲

بله. اینجا: (^)



نویسنده: Vb\_asp\_net  
تاریخ: ۱۱:۴۵:۳۶ ۱۳۹۰/۱۰/۲۲

استاد براتون مقدور هست من یک Sample کوچک براتون Email کنم که شما کدهای من رو نگاه کنید؟؟؟ چون واقعاً این مشکل واسه تمام پروژه هام خیلی حیاتی شده.

نویسنده: وحید نصیری  
تاریخ: ۱۳:۲۷:۴۴ ۱۳۹۰/۱۰/۲۲

سلام؛ بفرست.

نویسنده: Mohsen  
تاریخ: ۱۱:۰۶:۲۱ ۱۳۹۰/۱۰/۲۵

نکات واقعا ارزشمندی بود مهندس نصیری. ممنون از شما.