عنوان: **اتصال به بانک اراکل بدون نیاز به نصب oracleclient و یا استفاده از کنترلهای Devart**
نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۹:۵۷
آدرس: www.dotnettips.info
گروه‌ها: Oracle, VB.NET

گام اول: دانلود پکیج odp.net از وبسایت اراکل(Free). دانلود
گام دوم: فایلهای

filename	size
oci.dll	922kb
Oracle.DataAccess.dll	1,386kb
orannzsbb11.dll	1,256kb
oraociei11.dll	127,196kb
OraOps11w.dll	344kb

در کنار app کپی کنید

گام سوم: ایجاد connectionstring به صورت زیر:

```
Dim oradb As String = "Data Source=(DESCRIPTION=(ADDRESS_LIST=" _
                  + "(ADDRESS=(PROTOCOL=TCP)(HOST=YourServer)(PORT=YourPort)))" _
                  + "(CONNECT_DATA=(SERVER=DEDICATED)(SID=ORCL)));" _
                  + "User Id=UserName;Password=Password;"
        Dim conn As New OracleConnection(oradb)
        Try
            conn.Open()
        Catch ex As Exception
            Throw
        End Try
```

موفق باشید

نظرات خوانندگان

نویسنده: سجاد
تاریخ: ۱۳۹۲/۰۱/۲۴ ۹:۵

سلام
من از database first استفاده می‌کنم . می‌خوام از طریق edmx به دیتابیس اراکل وصل شم ولی پروایدر اراکل رو تو لیست پروایدرهای entity model نشون نمیده . حتی خواستم odac.net رو واسه اراکل دانلود کنم ولی همه لینکها ... . اگه ممکنه راهنمایی بفرمایید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۴ ۱۰:۱۷

دوست عزیز، اینقدر ضعیف برخورد نکن! دسترسی نداری؟ مهم نیست! خیلی از سایت‌ها هستند که امکان ریموت آپلود به شما می‌دن. مثلا می‌ری یک اکانت هات فایل یا رپیدشیر می‌خری، بعد نام یک قسمتش هست remote download. کاری که این قابلیت انجام می‌ده، دریافت فایل مورد نظر شما است بر اساس لینکی که بهش می‌دی با استفاده از IP واقعی سرور خودش.
نمونه همین کار رو وطنی‌ها هم انجام دادن. احتمالا دیدی یک سری از سایت‌ها اکانت‌های فایل‌های به اشتراک گذاری رو به اشتراک می‌گذارند! (پر شدن این‌ها؛ یک نوع کار آفرینی است! سایت‌های به اشتراک گذاری رو مسدود می‌کنند، برای این‌ها ایجاد اشتغال میشه!)
اکثر این‌ها هم قسمت دریافت مستقیم فایل رو دارند و نکته مهمی که در اینجا وجود دارد، دریافت فایل با IP سرور خارج از ایران است و نه با IP اصلی شما.

نویسنده: سجاد
تاریخ: ۱۳۹۲/۰۱/۳۰ ۲۰:۱۵

سلام به همه
odac.net رو دانلود کردم
ولی وقتی میخوام کانکشن edmx رو بسازم خطای could not resolve the connect identifier specified رو میده لطفا راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۳۰ ۲۰:۳۰

از جستجوی گوگل استفاده کنید: اولین لینک نتایج جستجو، توضیح رسمی خود اوراکل است در مورد خطایی که عنوان کردید.

نویسنده: سجاد
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۱:۴۶

باید ORACLE_HOME و TNS_ADMIN رو به لیست Environment Variables سیستم اضافه می‌کردم
با تشکر جناب نصیری

نویسنده: vahid
تاریخ: ۱۳۹۲/۱۲/۱۱ ۱۰:۴۰

با سلام بنده ODAC1120320_x64 را از سایت اوراکل گرفتم و نصب کردم ولی در زمان ایجاد edmx در لیست providerها ش نیست حتی در قسمت other هم همینطور علت خاصی دارد؟ ممنون

عنوان: **lambda expression در Vb.net**
نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۱:۱۱
آدرس: www.dotnettips.info
برچسب‌ها: Entity framework, VB.NET

با vb.net تو پروژه vs2012 خیلی بیرحمانه رفتار شد. خیلی از برنامه نویسان هنوز فکر میکنن vb.net زبان آموزشه نه برنامه نویسی. حالا با اومدن MVC, EF4, ParallelPrograming, Async و غیره موضوع بدتر شده و در اکثر وبسایتها معلولا بحث و sampleها با زبان C# ارائه میشه. اکثر Pattern که نوشته میشه به زبان C#. تو MVC4 بوضوع میتونید اینو لمس کنید.
امروز میخوام چندتا مثال از دستورات EF با *lambda expression* بنویسم.
دستور Select

```
Dim query = db.table.Select(Function(q)  q)
```

Select سفارشی

```
Dim query = db.table.Select(Function(q) New With{q.field1,q.field2,...})
```

Select پردن کردن فیلدهای یک Property
class:

```
Public Class person
Public Property id As Integet
Public Property fname As String
End Class
```

entity commmand:

```
Dim query = db.table.Where(Function(q) New person With{.id=q.idfield,.fname=q.namefield})
```

دستور where، single

```
Dim query = db.table.Where(Function(q) q.yourfield = yourvaribale).Select(Function(p) p).Single()
```

دستور join و group
join:

```
Dim query = db.table1.Join(db.table2, Function(q) q.youridfield, Function(p) p.youridfield, Function(q,
p) New With {Key.q= q, Key.p= p})
```

group:

```
Dim query = db.table.GroupBy(Function(q) q.groupkey).Select(Function(p) p.Sum(Function(w) w.aggregate))
```

مابقی دستورات مسه iinsert,delete,update نگذاشتم. اگر دوستان مایل بودن pm بدن این دستوراتم براشون میزارم
موفق باشید

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۵:۱۱

بی احترامی به دوستان VB کار نباشه ولی واقعاً VB زبان مزخرفی هستش و عمرش رو به پایان. C# هم یادگیریش خیلی راحت تره هم امکاناتش خیلی بیشتر. به نظرم دیگه از VB برای آموزش هم نباید استفاده کرد با توجه به این syntax بدی که داره.

نویسنده: ایمان محمدی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۶:۵۹

چه اصراری است از زبانی استفاده کنیم که مدرسان ، حرفه ای‌ها ، اپن سورس نویس ها، طراحان الگو و .... از اون استفاده نمی‌کنند. نمی‌خوام وارد بحث مسخره vb.net بهتر است یا C# بشم فقط اینقدر بدونید بخاطر انتخاب اشتباه زبان که خودم در شروع اون نقشی نداشتم دو سه سال از عمرم پای vb.net حروم شد و دوست ندارم *علاقمند واقعی برنامه نویسی* این اشتباه رو انجام بدن.
اگه به بحث‌های ساختاری این دو زبان کاری نداشته باشیم اینکه فضای کار و آموزش از vb.net استفاده نمی‌کنه بهترین دلیل برای دوری جست از این زبان برنامه نویسه.

کاشکی یک بند دیگه هم به وضعیت فناوری‌های مرتبط با دات نت از دیدگاه مرگ و زندگی! اضافه می‌شد و تازه کارها رو از vb.net نهی می‌کرد.
ببخشید که این نظر به محتوای فنی پست ارتباطی نداره.

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۷:۲۳

سلام البته این نظر شخصیتون بود.
درسته؟!
در ضمن سویچ کردن از VB به CSharp زمان زیادی نمی‌خواد و یا برعکس

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۷:۳۶

واقعا که اینم از اون حرفها بود
ویژگی برتر VB به Csharp تو راحت بودن یادگیریشه
چون به زبان محاوره نزدیکتره
امکاناتش خیلی بیشتر؟
جالب بود!
بازم نظر شخصی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۷:۴۷

البته من با VB.NET کار نمی‌کنم ولی بیشتر سلیقه‌ای است. معمار ارشد CLR آقای Anders Hejlsberg (ایشون فقط خالق سی شارپ یا پیشتر دلفی نیست؛ یکی از معماران ارشد CLR هم هست)، یکی از تلاش‌هاش هماهنگ کردن تمام زبان‌های رسمی دات نتی و یکپارچگی و سازگاری بین آن‌ها است. خلاصه هرکاری با یکی بتونی انجام بدی با بقیه هم می‌شود.
به علاوه در دلفی دات نت، syntax تعریف lambda expressions خیلی شبیه به VB.NET است.

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۷:۵۶

این هماهنگی هم داره بیشتر میشه
مثل اضافه کردن Iteratorها در vb.net11
تنها مورد بدرد بخوری که (من) ندیدم در vb.net ویژگی type forwarding بود

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۸:۲۰

یعنی واقعاً به نظر شما VB نسبت به C# به زبان محاوره نزدیک تره؟!
خواهشاً همین کدی که بالا نوشته شده رو بخونید ببینید آیا به زبان محاوره نزدیک هست یا نه؟

یکی از سوالایی که همیشه تو ذهن من وجود داشته این بوده که چرا برای آموزش از VB استفاده میکنن؟!

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۸:۴۳

می‌دونی چرا می‌گن به محاوره نزدیکتره؟
چون تو vb.net برای مشتق گیری از کلمه کلیدی Inherits استفاده میشه که معنیش مشخصه اما تو CSharp میشه :
برای پیاده سازی اینترفس‌ها ازکلمه کیدی Implements استفاده میشه که معنیش میشه پیاده سازی ولی تو Csharp میشه بازم :
و.....
منظور از این که به زبان محاوره نزدیکه بخاطر کلمات کلیدیش هست
نه اینکه بخوای با جمله‌بندی کدنویسی کنی
فکر می‌کنم دیگه این سوال رو باید از ذهنتون پاک کنید

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۲۰ ۱:۳

What's New for Visual Basic in Visual Studio 2012 RC

نویسنده: آرمان
تاریخ: ۱۳۹۱/۰۷/۰۴ ۱:۵۶

اصولا این گونه نظرات نسبت به زبانی مثل وی بی دات نت به دلیل عدم سابقه تجربی و دانش کافی در مورد آن است. از نظر کارایی و سرعت و قدرت که تفاوت خاصی بین زبان‌های قابل استفاده در دات نت نیست. از نظر زیبایی سینتکس وی بی دات نت برتری داره و به همین دلیل برای آموزش و شروع کار بسیار بهتر است و برای ادامه هم هیچ مشکلی ندارد. سی شارپ هم جذابیت‌های خود را دارد. گاهی همان خلاصه نویسی آن لذت بخش است. اما اصولا یه برنامه نویس حرفه ای که یکی از این زبان‌ها را انتخاب کرده به راحتی می‌تواند در چند ساعت در دیگری نیز مهارت لازم را کسب نماید. بحث عمر تلف شدن در وی بی بسیار جالب است!

عنوان: **ایجاد یک Repository در پروژه برای دستورات EF**
نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۱:۴۷
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Design patterns, Entity framework, VB.NET

ایجاد یک Pattern در پروژتون میتونه نظم, سرعت و زیبایی خاصی به کدتون بده. با وجود frameworkهای و Patternهایی مسه MVC و MVVM برنامه نویسان را وادار کنه که همه Actionهای یک پروژه رو به سمت کلاینت ببرن.تو یک فرصت دیگه در مورد فریمورک Knockout حتما تاپیک میزارم.امروز میخوام یک Pattern با استفاده از یک Interface و codefirst model براتون بزارم.

گام اول:ایجاد که class property

```
Public Class Employee
    Public Property ID As Integer
    Public Property Fname As String
    Public Property Bdate As DateTime
End Class
```

گام دوم:ایجاد بانک با استفاده از CodeFirst

```
Imports System.Data.Entity
Public Class EmployeeDBContext : Inherits DBContext
    Public Property Employees As DbSet(Of Employee)
End Class
```

گام سوم: ایجاد repository با استفاده از interface

```
Interface EmployeeRepository
    ReadOnly Property All As List(Of Employee)
    Function Find(id As Integer) As Employee
    Sub InsertOrUpdate(p As Employee)
    Sub Delete(id As Integer)
    Sub Save()
End Interface
```

گام چهارم: تعریف کلاس برای implement کردن از iInterface

```
Public Class EmployeeClass : Implements EmployeeRepository
    Private DB As New EmployeeDBContext
    Public ReadOnly Property All As List(Of Employee) Implements EmployeeRepository.All
        Get
            Return DB.Employees.ToList()
        End Get
    End Property

    Public Sub Delete(id As Integer) Implements EmployeeRepository.Delete
        Dim query = DB.Employees.Single(Function(q) q.ID = id)
        DB.Employees.Remove(query)
    End Sub

    Public Function Find(id As Integer) As Employee Implements EmployeeRepository.Find
        Return DB.Employees.Where(Function(q) q.ID = id)
    End Function

    Public Sub InsertOrUpdate(p As Employee) Implements EmployeeRepository.InsertOrUpdate
        If p.ID = Nothing Then
            DB.Employees.Add(p)
        Else
            DB.Entry(p).State = Data.EntityState.Modified
        End If
    End Sub

    Public Sub Save() Implements EmployeeRepository.Save
        DB.SaveChanges()
    End Sub
```

```
End Class
```

برای استفاده تو پروژه براحتی میتونید یک instance از classتون ایجاد کنید و ..

```
Dim cls As New EmployeeClass
```

```
Public Sub BindGrid()
        GridView1.DataSource = cls.All
        GridView1.DataBind()
    End Sub
```

موفق باشید

نظرات خوانندگان

نویسنده: علیرضا صالحی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۱۳

برای مواردی که خروجی یک لیست (تعدادی آیتم) باشد از Property استفاده نمی‌شود. مثلا برای All باید از Method استفاده کنید. Properties vs. Methods
بهتر است برای خروجی متد هایی مانند All نیز به جای لیست از IEnumerable یا IQueryable استفاده کنید.
متدهای Update و Insert نیز به طور جداگانه تعریف شوند. (قرار است هر متد تنها یک وظیفه داشته باشد)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۲۱

- در مورد آرایه بحث شده در MSDN. ضمن اینکه استفاده از متد عموما برای حالتیکه عملیات قابل توجهی در بدنه آن قرار است صورت گیرد، توصیه می‌شود . البته در اینجا چون عملیات دریافت اطلاعات از بانک اطلاعاتی می‌تواند سنگین درنظر گرفته شود، استفاده از متد ارجحیت دارد. خواص نمایانگر اطلاعاتی سبک و با دسترسی سریع هستند.
- خروجی لیست بهتر است. (^) + اگر ReSharper جدید را نصب کنید استفاده از IEnumerable را نیز توصیه نمی‌کند ؛ چون ممکن است چندین بار رفت و برگشت به بانک اطلاعاتی در این بین صورت گیرد.
- مشکلی ندارد. خود EF Code first چنین متدی را دارد. (^) بحث کلاس تک وظیفه‌ای متفاوت است با متدی که نهایتا قرار است اطلاعات یک رکورد را در بانک اطلاعاتی تغییر دهد (اگر نبود ثبتش کند؛ اگر بود فقط همان رکورد مشخص را به روز رسانی کند).

نویسنده: ناشناس
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۳۷

با سلام
دلیل استفاده از Interface EmployeeRepository چیه؟
دقیقا دلیل استفاده Interface اینجا چیه؟
با تشکر از مطلب خوبتون.

نویسنده: ناشناس
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۴۵

اینجا شاید استفاده از IQueryable بهتر باشه.
شاید کاربر بخواد قبل از نمایش اطلاعات اونو فیلتر کنه یا اینکه بهتره دو متد Find داشته باشی یکی با خروجی یک آیتم و دیگری با خروجی چندین آیتم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۴۸

خیر (^). طراحی یک لایه سرویس که خروجی IQueryable دارد نشتی دار درنظر گرفته شده و توصیه نمی‌شود. اصطلاحا leaky abstraction هم به آن گفته می‌شود؛ چون طراح نتوانسته حد و مرز سیستم خودش را مشخص کند و همچنین نتوانسته سازوکار درونی آن‌را به خوبی کپسوله سازی و مخفی نماید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۵۰

به دو دلیل:
- استفاده از امکان تزریق وابستگی‌ها

- امکان نوشتن ساده‌تر آزمون‌های واحد با فراهم شدن زیر ساخت mocking اشیاء

نویسنده: ناشناس
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۳:۲

در همین پست خود شما تعداد زیاد رکوردها رو مثال زدیدو این پیاده سازی از این موضوع رنج میبره. و در مورد IQueryable قبول دارم و گفتم که بهتر است از دو یا چند متد find استفاده کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۳:۲۵

منظور از آن مطلب این بود که از ابزاری که در اختیار دارید درست استفاده کنید. اگر قرار است دو یا چند جستجو را انجام دهید، اینکارها بله باید با IQueryable داخل یک متد انجام شود، اما خروجی متد فقط باید لیست حاصل باشد؛ نه IQueryable ایی که انتهای آن باز است و سبب نشتی لایه سرویس شما در لایه‌های دیگر خواهد شد.

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۹:۵۴

برای برنامه نویسا پیدا کردن یک property راحتره در ضمن از property برای تزریق یا بازیابی اطلاعات از یک object استفاده میکنن.
IQueryable در واقع توسعه یافته IEnumerable.تفاوت عمدشون در LINQ operators که در IQueryable استفاده میشه.اگر هم بخوایم دلیل پیشنهادی داده باشیم اونم اینه که مدیریت حافظه در IQueryable رعایت شده در حالی که Lisrtها کامپایلرو مجاب به اجنام دستور تا انتها میکنن.

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۹:۵۹

مهندس با نظر دوستمون موافقم
IQueryable بهترین انتخاب برای remote data source که میشه به database یا webserviceها اشاره کرد.بطور کل اگر شما از ORM مسه linqtosql استفاده میکنید
IQueriable : کوئری شمارو به دستورات sql در database server تبدیل میکنه
IEnumerable: همه رکوردهای شما قبل از اینکه بسمت دیتابیس برن بصورت object در memory نگهداری میشن.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۲۰:۱۱

IQueryable فقط یک expression است. هنوز اجرا نشده. (expose آن از طریق وب سرویس اشتباه است و به مشکلات serialization برخواهید خورد.)
زمانیکه First، ToList و امثال آن روی این عبارت فراخوانی شود تبدیل به SQL شده و سپس بر روی بانک اطلاعاتی اجرا می‌شود. به این deferred execution یا اجرای به تعویق افتاده گفته می‌شود.
اگر این عبارت را در اختیار لایه‌های دیگر قرار دهید، یعنی انتهای کار را بازگذاشته‌اید و حد و حدود سیستم شما مشخص نیست. شما اگر IQueryable بازگشت دهید، در لایه‌ای دیگر می‌شود یک join روی آن نوشت و اطلاعات چندین جدول دیگر را استخراج کرد؛ درحالیکه نام متد شما GetUsers بوده. بنابراین بهتر است به صورت صریح اطلاعات را به شکل List بازگشت دهید، تا انتهای کار باز نمانده و طراحی شما نشتی نداشته باشد.

نویسنده: محمد عامریان
تاریخ: ۱۳۹۱/۰۸/۱۸ ۱۷:۶

با سلام من یک معماری طراحی کردم به شکل زیر
ابتدا یک اینترفیس به شکل زیر دارم

```
using System;
using System.Collections;
using System.Linq;

namespace Framework.Model
{
    public interface IContext
    {
        T Get<T>(Func<T, bool> prediction) where T : class;
        IEnumerable List<T>(Func<T, bool> prediction) where T : class;
        void Insert<T>(T entity) where T : class;
        int Save();
    }
}
```

بعد یک کلاس ازش مشتق شده

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Text;

namespace Framework.Model
{
    public class Context : IContext
    {
        private readonly DbContext _dbContext;

        public Context(DbContext context)
        {
            _dbContext = context;
        }

        public T Get<T>(Func<T,bool> prediction) where T : class
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet!= null)
                return dbSet.Single(prediction);

            throw new Exception();
        }

        public void Insert<T>(T entity) where T : class
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet != null)
            {
                _dbContext.Entry(entity).State = EntityState.Added;
            }
        }

        public int Save()
        {
            return _dbContext.SaveChanges();
        }


        IEnumerable IContext.List<T>(Func<T, bool> prediction)
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet != null)
                return dbSet.Where(prediction).ToList();

            throw new Exception();
        }
    }
}
```

سپس یک کلاش context دارم که مستقیما از dbcontext مشتق شده

```
using System.Data.Entity;
using DataModel;

namespace Model
{
    public class EFContext : DbContext
    {
        public EFContext(string db): base(db)
        {

        }

        public DbSet<Product> Products { get; set; }
    }
}
```

و سپس کلاس دارم که اومده پیاده سازی کرده context که خودم ساختمو

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;

namespace Model
{
    public class Context : Framework.Model.Context
    {
        public Context(string db): base(new EFContext(db))
        {

        }
    }
}
```

در پروژه دیگری اومدم یک کلاس context جدید ساختم

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Biz
{
    public class Context : Model.Context
    {
        public Context(string db) : base(db)
        {

        }
    }
}
```

و در کنترلر هم به این شکل ازش استفاده کردم

```
using System.Web.Mvc;
using Framework.Model;

namespace ProductionRepository.Controllers
{
    public class BaseController : Controller
    {
        public IContext DataContext { get; set; }

        public BaseController()
        {
            DataContext = new
Biz.Context(System.Configuration.ConfigurationManager.ConnectionStrings["Database"].ConnectionString);
        }
    }
}
```

```
using System.Web.Mvc;
using DataModel;
using System.Collections.Generic;

namespace ProductionRepository.Controllers
{
    public class ProductController : BaseController
    {
        public ActionResult Index()
        {
            var x = DataContext.List<Product>(s => s.Name != null);
            return View(x);
        }

    }
}
```

و این هم تست

```
using NUnit.Framework;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Mvc;

namespace TestUnit
{
    [TestFixture]
    public class Test
    {
        [Test]
        public void IndexShouldListProduct()
        {
            var repo = new Moq.Mock<Framework.Model.IContext>();
            var products = new List<DataModel.Product>();
            products.Add(new DataModel.Product { Id = 1, Name = "asdasdasd" });
            products.Add(new DataModel.Product { Id = 2, Name = "adaqwe" });
            products.Add(new DataModel.Product { Id = 4, Name = "qewqw" });
            products.Add(new DataModel.Product { Id = 5, Name = "qwe" });
            repo.Setup(x => x.List<DataModel.Product>(p => p.Name !=
null)).Returns(products.AsEnumerable());
            var controller = new ProductionRepository.Controllers.ProductController();
            controller.DataContext = repo.Object;
            var result = controller.Index() as ViewResult;
            var model = result.Model as List<DataModel.Product>;
            Assert.AreEqual(4, model.Count);
            Assert.AreEqual("", result.ViewName);

        }
    }
}
```

نظرتون چیه آقای نصیری

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۸ ۱۷:۱۳

موارد 1 و 2 عنوان شده در این مطلب رو تکرار کرده: (^)

نویسنده: مجید پارسا
تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۲:۹

با سلام؛ سوالی که وجود داره اینه که با استفاده از repository pattern چطور میتونیم join بزنیم. با توجه به نظرات قبلی توصیه شده است که از خروجی IQueryable نباید برای لایه داده استفاده شود.
در این صورت در هنگام نوشتن دستورات join ابتدا تمامی رکوردهای جداول مورد نظر توسط الگوی repository به حافظه load می‌شود، با توجه به ماهیت linq to object بودن کوئری مورد نظر(join) اجرای برنامه به لحاظ زمانی و مصرف حافظه از

کارایی خوبی برخوردار نخواهد بود.
در این حالت یا می‌باست از خیر کارایی بالاتر گذشت یا از خروجی IQueryable استفاده کرد که در تضاد با پیشنهاد دوستان گرامی می‌باشد.
آیا در این حالت منطقی است joinهای پر استفاده را با خروجی IEnumerable در repository مربوط به خودش نوشت یا راهکار دیگری وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۲:۱۹

- الگوی مخزن عمومی (Generic repository pattern)، لایه داده برنامه نیست. زمانیکه از یک ORM استفاده می‌کنید، لایه داده برنامه همان ORM است.
- الگوی مخزن عمومی، عمده‌ی کارش مخفی کردن ساز و کار ORM مورد استفاده از لایه سرویس برنامه است (^).
- اگر از الگوی عمومی مخزن استفاده می‌کنید، سطح دسترسی آن‌را internal تعریف کنید تا محدود شود به لایه سرویس برنامه. داخل لایه سرویس برنامه به هر نحوی که علاقمندید از آن استفاده کنید. نهایتا این لایه سرویس است که خروجی IList یا IEnumerable نهایی را در اختیار مصرف کننده قرار می‌دهد.

نویسنده: مجید پارسا
تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۶:۸

با تشکر، از آنجا که من اولین بار است که به شکل حرفه‌ای برنامه نویسی سه لایه را تجربه می‌کنم با توجه به توضیحات شما این طور متوجه شدم که پیاده سازی کلاس‌های Repository در لایه سرویس صورت گیرد اگر اشتباه نکنم.

در صورت امکان بیشتر موضوع رو باز کنید (منظورم آماتوری تره)
نمونه برنامه‌های سه لایه موجود در اینترنت پیدا کردم در حد CRUD ساده و با استفاده از الگوی مخزن عمومی بوده. مانند مثال‌های سایت asp.net در صورت معرفی نمونه کاملتر و واقعی‌تر ممنون میشوم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۷:۲۹

مراجعه کنید به مسیر راه EF Code first ، انتهای مطلب، قسمت لایه بندی پروژه‌های EF Code first

عنوان: Parallel Programming در Vb.Net
نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۱۴
آدرس: www.dotnettips.info
برچسب‌ها: LINQ, Task Programming Language, Parallel Extensions, VB.NET

حقیقتا تا این لحظه تو پروژه ای استفاده نکردم ولی فکر میکنم یادگیری و استفادش ضروری باشه. ظهورش برمیگرده به .net1 با عنوان Threading. اما کار با Threading خیلی مشکله. من که اینطوری فکر میکنم. حالا با اصلاح کلاس Threading و آمده task خیلی بهتر شده.

گام اول:Threading.Tasks را بعنوان namespace اضافه کنید

یک مثال: این loop در نظر بگیرید

```
Private Sub work()
        While True
        End While
End Sub
```

میخوام برا متد بالا یک task تعریف کنم

```
Task.Factory.StartNew(Sub() work())
```

مثال دوم: یک لیست تعریف میکنم و با استفاد از یک loop میخوام اجزا لیستو چاپ کنم.

```
Dim lst As New List(Of String) From {"meysam", ".nettips", "vahidnasirii"}
Parallel.ForEach(lst, Sub(item) Console.WriteLine("name:{0}", item))
```

مثال سوم: میخوام از این تکنیک تو linq استفاده کنیم:

```
Dim no(9) As Integer
   For i As Integer = 0 To no.Length - 1
        no(i) = i
   Next
Dim result As IEnumerable(Of Double) = no.AsParallel.Select(Function(q) Math.Pow(q,
3)).OrderBy(Function(q) q)
   For Each items In result
        Console.WriteLine(items)
   Next
```

موفق باشید.

نظرات خوانندگان

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۳:۹

سلام
این کد از لحاظ منطقی درسته و جواب می‌ده ولی کاملا اشتباست چون sub رو بی‌دلیل نوشتی

```
Task.Factory.StartNew(Sub() work())
        'نحوه‌ی صحیح نوشتنش

        Task.Factory.StartNew(AddressOf work)
        '---- یا
        Task.Factory.StartNew(Sub()
                                  While True
                                  End While
                              End Sub)
```

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۹:۳۵

AddressOf در دستور Threading که قدیمی هست استفاده میشه که عمدتا بصورت:

```
Dim t As New Threading.Thread(AddressOf work)
t.Start()
```

متد Work برای این تعریف شده که مفوم کد برسونه.

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۹:۴۶

سلام میثم جان اشتباه نکن

AddressOf

ربطی به Thread و یا Task نداره

از AddressOf برای ارجاع به Procedure و Functionها استفاده میشه

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۱/۰۵/۲۰ ۱:۱۲

دوست من منظور من این نبود که AddressOf ارتباطی با Threading داره. منظور من این بود که از زمانی که من Parallel Programmimg کار کردم جایی ندیدم از AddressOf تو دستور Task یا Parallel استفاده کنن.از این دستور تو Thread یا BackgroundWorking استفاده میشد که نسبتا تو نسخه‌های قدیمی .net هستن.