عنوان: **آشنایی با NHibernate - قسمت اول**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۷/۱۷ ۲۱:۱۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: NHibernate

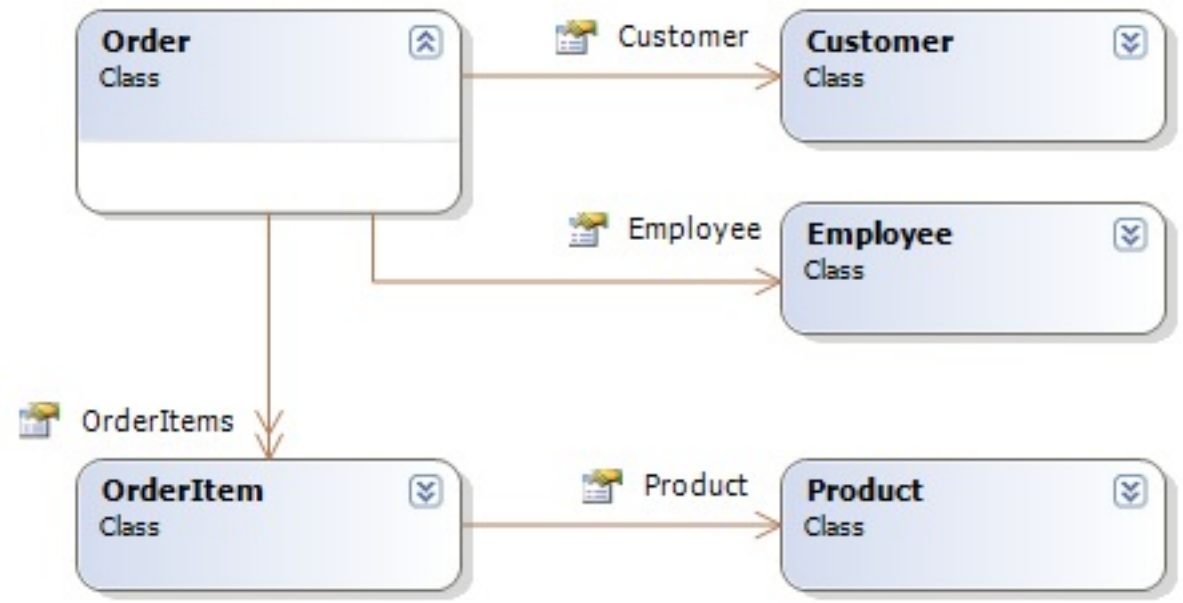
NHibernate کتابخانه‌ی تبدیل شده پروژه بسیار محبوب Hibernate جاوا به سی شارپ است و یکی از ORM های بسیار موفق، به شمار می‌رود. در طی تعدادی مقاله قصد آشنایی با این فریم ورک را داریم.

چرا نیاز است تا از یک ORM استفاده شود؟
تهیه قسمت و یا لایه دسترسی به داده‌ها در یک برنامه عموما تا 30 درصد زمان کل تهیه یک محصول را تشکیل می‌دهد. اما باید در نظر داشت که این پروسه‌ی تکراری هیچ کار خارق العاده‌ای نبوده و ارزش افزوده‌ی خاصی را به یک برنامه اضافه نمی‌کند.
تقریبا تمام برنامه‌های تجاری نیاز به لایه دسترسی به داده‌ها را دارند. پس چرا ما باید به ازای هر پروژه، این کار تکراری و کسل کننده را بارها و بارها تکرار کنیم؟
هدف NHibernate ، کاستن این بار از روی شانه‌های یک برنامه نویس است. با کمک این کتابخانه، دیگر رویه ذخیره شده‌ای را نخواهید نوشت. دیگر هیچگاه با ADO.Net سر و کار نخواهید داشت. به این صورت می‌توان عمده وقت خود را صرف قسمت‌های اصلی و طراحی برنامه کرد تا کد نویسی یک لایه تکراری. همچنین عده‌ای از بزرگان اینگونه ابزارها اعتقاد دارند که برنامه نویس‌هایی که لایه دسترسی به داده‌ها را خود طراحی می‌کنند، مشغول کلاهبرداری از مشتری‌های خود هستند! (صرف زمان بیشتر برای تهیه یک محصول و همچنین وجود باگ‌های احتمالی در لایه دسترسی به داده‌های طراحی شده توسط یک برنامه نویس نه چندان حرفه‌ای)
برای مشاهده سایر مزایای استفاده از یک ORM لطفا به مقاله " 5 دلیل برای استفاده از یک ابزار ORM " مراجعه نمائید.

در ادامه برای معرفی این کتابخانه یک سیستم ثبت سفارشات را با هم مرور خواهیم کرد.

بررسی مدل سیستم ثبت سفارشات

در این مدل ساده‌ی ما، مشتری‌ها (customers) امکان ثبت سفارشات (orders) را دارند. سفارشات توسط یک کارمند (employee) که مسؤول ثبت آن‌ها است به سیستم وارد می‌شود. هر سفارش می‌تواند شامل یک یا چند (one-to-many) آیتم (order items) باشد و هر آیتم معرف یک محصول (product) است که قرار است توسط یک مشتری (customer) خریداری شود.
کلاس دیاگرام این مدل به صورت زیر می‌تواند باشد.



نگاشت مدل

زمانیکه مدل سیستم مشخص شد، اکنون نیاز است تا حالات (داده‌ها) آن‌را در مکانی ذخیره کنیم. عموما اینکار با کمک سیستم‌های مدیریت پایگاه‌های داده مانند MySql ، IBM DB2، Oracle، SQL Server و امثال آن‌ها صورت می‌گیرد. زمانیکه از NHibernate استفاده کنید اهمیتی ندارد که برنامه شما قرار است با چه نوع دیتابیسی کار کند؛ زیرا این کتابخانه اکثر دیتابیس‌های شناخته شده موجود را پشتیبانی می‌کند و برنامه از این لحاظ مستقل از نوع دیتابیس عمل خواهد کرد و اگر نیاز بود روزی بجای اس کیوال سرور از مای اس کیوال استفاده شود، تنها کافی است تنظیمات ابتدایی NHibernate را تغییر دهید (بجای بازنویسی کل برنامه).

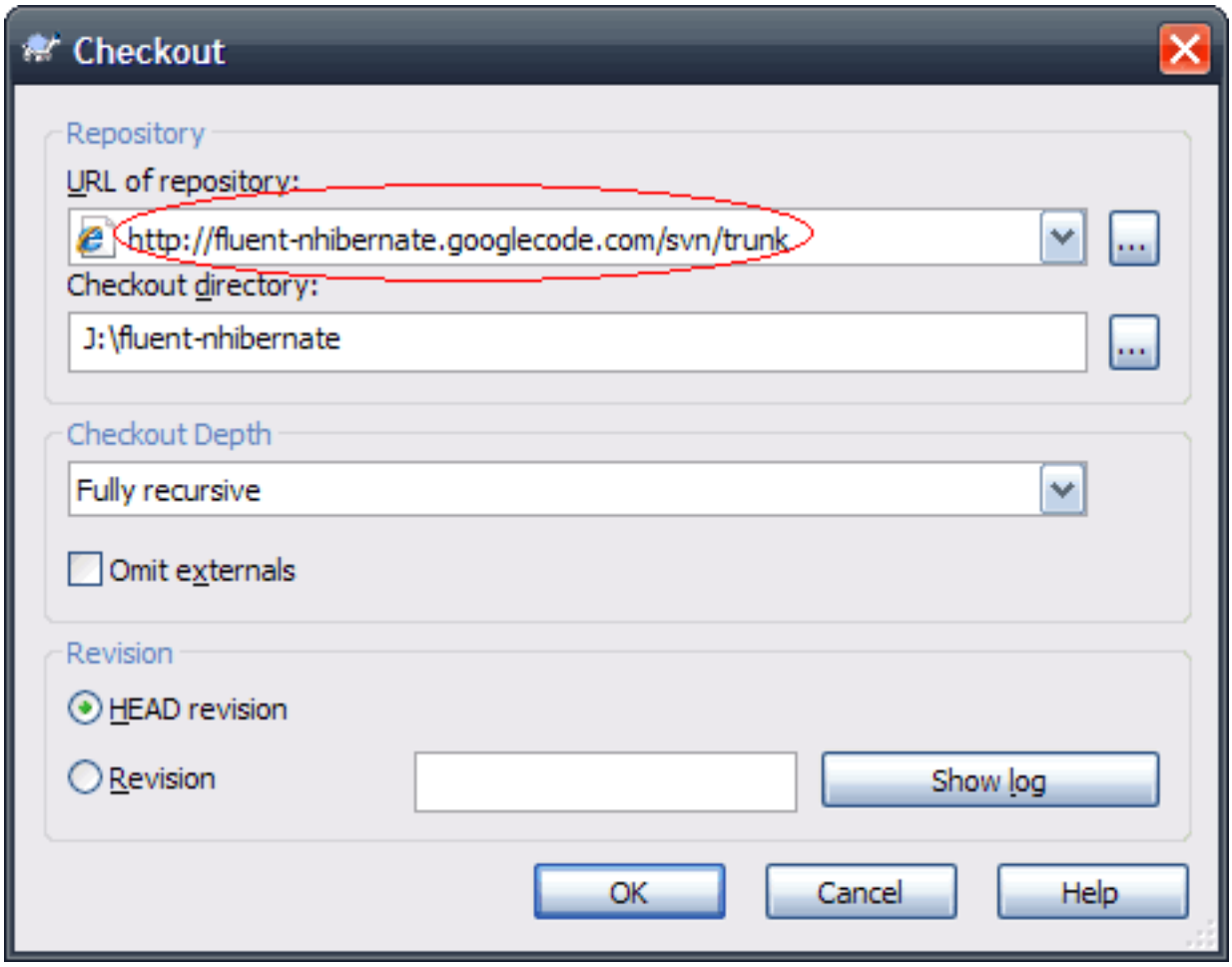
اگر برای ذخیره سازی داده‌ها و حالات سیستم از دیتابیس استفاده کنیم، نیاز است تا اشیاء مدل خود را به جداول دیتابیس نگاشت نمائیم. این نگاشت عموما یک به یک نیست (لزومی ندارد که حتما یک شیء به یک جدول نگاشت شود). در گذشته‌ی نچندان دور کتابخانه‌ی NHibernate ، این نگاشت عموما توسط فایل‌های XML ایی به نام hbm صورت می‌گرفت. این روش هنوز هم پشتیبانی شده و توسط بسیاری از برنامه نویس‌ها بکار گرفته می‌شود. روش دیگری که برای تعریف این نگاشت مرسوم است، مزین سازی اشیاء و خواص آن‌ها با یک سری از ویژگی‌ها می‌باشد که فریم ورک برتر این عملیات Castle Active Record نام دارد.

اخیرا کتابخانه‌ی دیگری برای انجام این نگاشت تهیه شده به نام Fluent NHibernate که بسیار مورد توجه علاقمندان به این فریم ورک واقع گردیده است. با کمک کتابخانه‌ی Fluent NHibernate عملیات نگاشت اشیاء به جداول، بجای استفاده از فایل‌های XML ، توسط کدهای برنامه صورت خواهند گرفت. این مورد مزایای بسیاری را همانند استفاده از یک زبان برنامه نویسی کامل برای تعریف نگاشت‌ها، بررسی خودکار نوع‌های داده‌ای و حتی امکان تعریف منطقی خاص برای قسمت نگاشت برنامه، به همراه خواهد داشت.

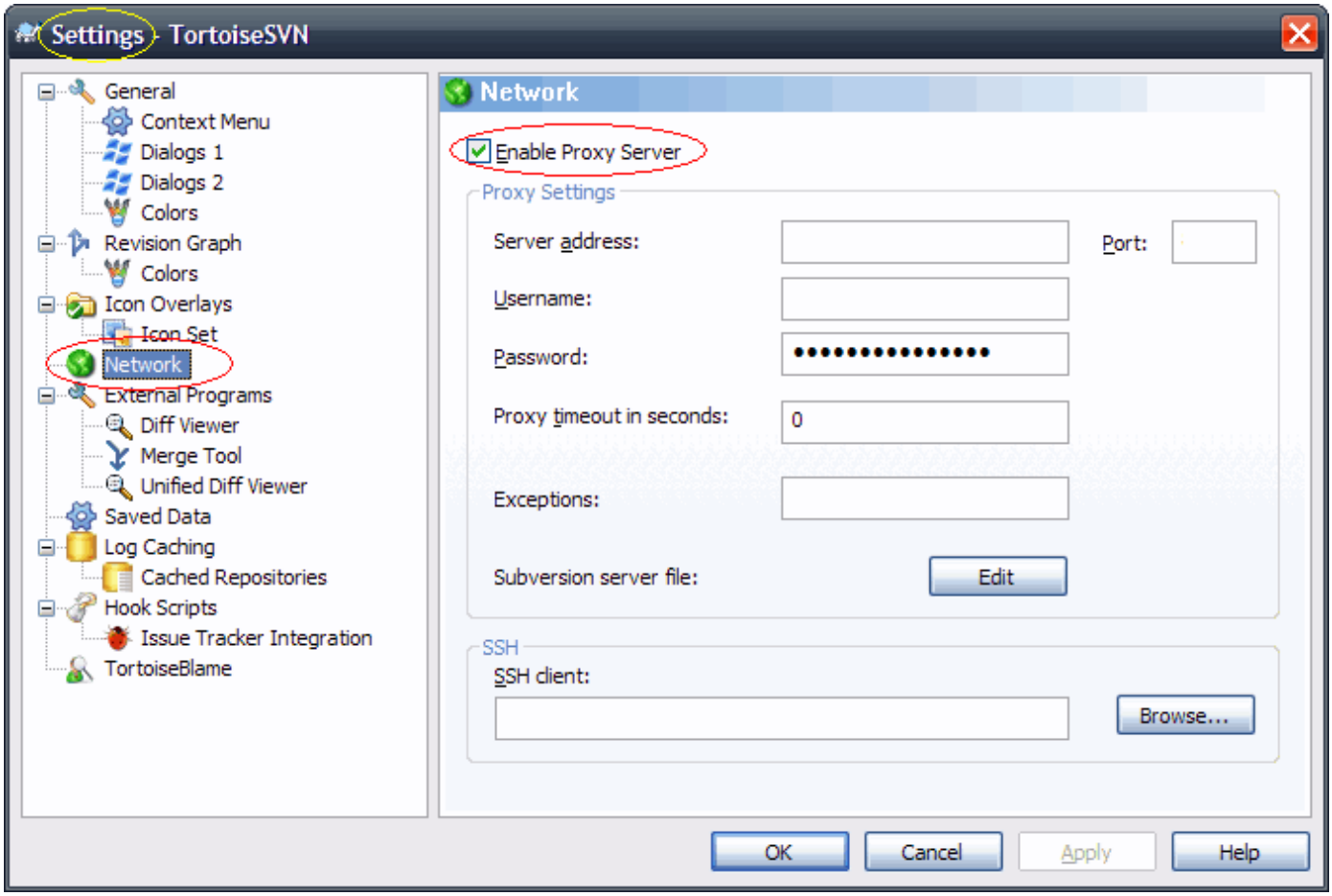
آماده سازی سیستم برای استفاده از NHibernate

در ادامه بجای دریافت پروژه سورس باز NHibernate از سایت سورس فورج، پروژه سورس باز Fluent NHibernate را از سایت گوگل کد دریافت خواهیم کرد که بر فراز کتابخانه‌ی NHibernate بنا شده است و آن‌را کاملا پوشش می‌دهد. سورس این کتابخانه را با checkout مسیر زیر توسط TortoiseSVN می‌توان دریافت کرد.

http://fluent-nhibernate.googlecode.com/svn/trunk



البته احتمالا برای دریافت آن از گوگل کد با توجه به تحریم موجود نیاز به پروکسی خواهد بود. برای تنظیم پروکسی در TortoiseSVN به قسمت تنظیمات آن مطابق تصویر ذیل مراجعه کنید:



همچنین جهت سهولت کار، آخرین نگارش موجود در زمان نگارش این مقاله را از این آدرس نیز می‌توانید دریافت نمائید.

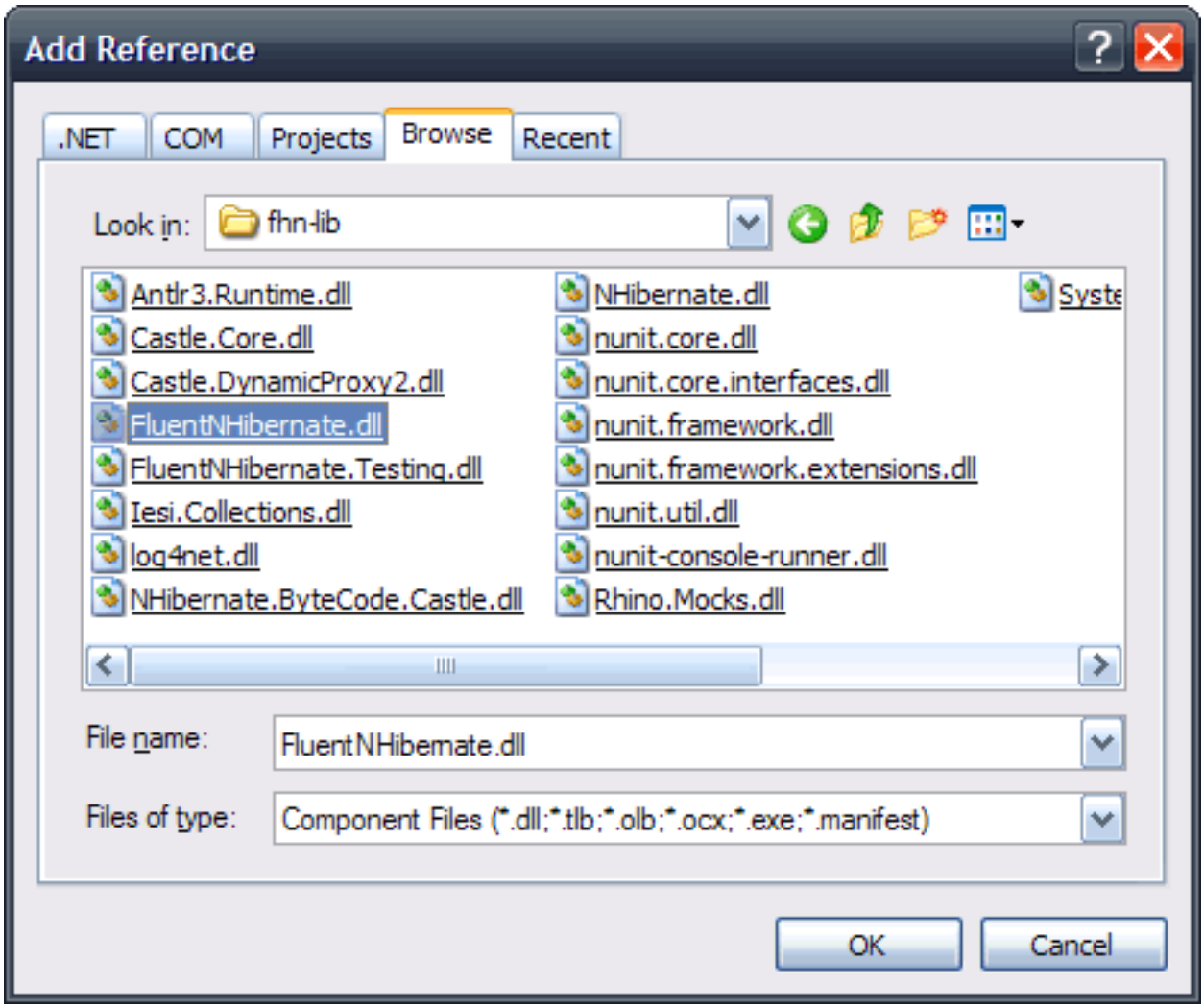
پس از دریافت پروژه، باز کردن فایل solution آن در VS و سپس build کل مجموعه، اگر به پوشه‌های آن مراجعه نمائید، فایل‌های زیر قابل مشاهده هستند:

Nhibernate.dll : اسمبلی فریم ورک NHibernate است.
NHibernate.Linq.dll : اسمبلی پروایدر LINQ to NHibernate می‌باشد.
FluentNHibernate.dll : اسمبلی فریم ورک Fluent NHibernate است.
Iesi.Collections.dll : یک سری مجموعه‌های ویژه مورد استفاده NHibernate را ارائه می‌دهد.
Log4net.dll : فریم ورک لاگ کردن اطلاعات NHibernate می‌باشد. (این فریم ورک نیز جهت عملیات logging بسیار معروف و محبوب است)
Castle.Core.dll : کتابخانه پایه Castle.DynamicProxy2.dll است.
Castle.DynamicProxy2.dll : جهت اعمال lazy loading در فریم ورک NHibernate بکار می‌رود.
System.Data.SQLite.dll : پروایدر دیتابیس SQLite است.
Nunit.framework.dll : نیز یکی از فریم ورک‌های بسیار محبوب آزمون واحد در دات نت فریم ورک است.

برای سادگی مراجعات بعدی، این فایل‌ها را یافته و در پوشه‌ای به نام lib کپی نمائید.

برپایی یک پروژه جدید

پس از دریافت Fluent NHibernate ، یک پروژه Class Library جدید را در VS.Net آغاز کنید (برای مثال به نام NHSample1).
سپس یک پروژه دیگر را نیز از نوع Class Library به نام UnitTests به این solution ایجاد شده جهت انجام آزمون‌های واحد برنامه اضافه نمائید.
اکنون به پروژه NHSample1 ، ارجاع هایی را به فایل‌های FluentNHibernate.dll و سپس NHibernate.dll در که پوشه lib ایی که در قسمت قبل ساختیم، قرار دارند، اضافه نمائید.



در ادامه یک پوشه جدید به پروژه NHSample1 به نام Domain اضافه کنید. سپس به این پوشه، کلاس Customer را اضافه نمائید:

```
namespace NHSample1.Domain
{
   public class Customer
   {
       public int Id { get; set; }
       public string FirstName { get; set; }
       public string LastName { get; set; }
       public string AddressLine1 { get; set; }
       public string AddressLine2 { get; set; }
       public string PostalCode { get; set; }
       public string City { get; set; }
       public string CountryCode { get; set; }
   }
}
```

اکنون نوبت تعریف نگاشت این شیء است. این کلاس باید از کلاس پایه ClassMap مشتق شود. سپس نگاشت‌ها در سازنده‌ی این کلاس باید تعریف گردند.

```
using FluentNHibernate.Mapping;

namespace NHSample1.Domain
{
   class CustomerMapping : ClassMap<Customer>
   {
   }
}
```

همانطور که ملاحظه می‌کنید، نوع این کلاس Generic ، همان کلاسی است که قصد داریم نگاشت مرتبط با آن را تهیه نمائیم. در ادامه تعریف کامل این کلاس نگاشت را در نظر بگیرید:

```
using FluentNHibernate.Mapping;

namespace NHSample1.Domain
{
   class CustomerMapping : ClassMap<Customer>
   {
       public CustomerMapping()
       {
           Not.LazyLoad();
           Id(c => c.Id).GeneratedBy.HiLo("1000");
           Map(c => c.FirstName).Not.Nullable().Length(50);
           Map(c => c.LastName).Not.Nullable().Length(50);
           Map(c => c.AddressLine1).Not.Nullable().Length(50);
           Map(c => c.AddressLine2).Length(50);
           Map(c => c.PostalCode).Not.Nullable().Length(10);
           Map(c => c.City).Not.Nullable().Length(50);
           Map(c => c.CountryCode).Not.Nullable().Length(2);
       }
   }
}
```

به صورت پیش فرض نگاشت‌های Fluent NHibernate از نوع lazy load هستند که در اینجا عکس آن در نظر گرفته شده است. سپس وضعیت نگاشت تک تک خواص کلاس Customer را مشخص می‌کنیم. توسط Id(c => c.Id).GeneratedBy.HiLo به سیستم اعلام خواهیم کرد که فیلد Id از نوع identity است که از 1000 شروع خواهد شد. مابقی موارد هم بسیار واضح هستند. تمامی خواص کلاس Customer ذکر شده، نال را نمی‌پذیرند (منهای AddressLine2) و طول آن‌ها نیز مشخص گردیده است.
با کمک Fluent NHibernate ، بحث بررسی نوع‌های داده‌ای و همچنین یکی بودن موارد مطرح شده در نگاشت با کلاس اصلی Customer به سادگی توسط کامپایلر بررسی شده و خطاهای آتی کاهش خواهند یافت.

برای آشنایی بیشتر با lambda expressions می‌توان به مقاله زیر مراجعه کرد:
Step-by-step Introduction to Delegates and Lambda Expressions

ادامه دارد...

نظرات خوانندگان

نویسنده: dadoo
تاریخ: ۱۳۸۸/۰۷/۱۸ ۰۸:۴۳:۳۰

آقای نصیری عزیز
باسلام
آیا استفاده از این ORM مناسبتر است یا LINQ؟ آیا می توانید مقایسه ای هر چند مختصر بین این دو ابزار داشته باشید. ممنون

نویسنده: LoveAjax
تاریخ: ۱۳۸۸/۰۷/۱۸ ۱۰:۱۲:۰۳

ایا nhibernate و fluent تحت هاست های medium trust اجرا می شوند

نویسنده: DotNetCoders
تاریخ: ۱۳۸۸/۰۷/۱۸ ۱۲:۲۰:۱۴

سلام!

جناب نصیری NHibernate رو میشه با VB.Net هم پیاده سازی کرد؟ یا فقط C# ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۷/۱۸ ۱۳:۱۳:۱۱

@DotNetCoders
سورس اصلی کتابخانه، به زبان سی شارپ است اما نهایتا شما از اسمبلی‌های کامپایل شده مربوطه استفاده خواهید کرد و از اینجا به بعد دیگر تفاوتی نمی‌کند که زبان دات نتی مورد استفاده چی باشد.

@dadoo
باید دقت داشته باشید که LINQ به تنهایی فقط یک language feature است و نه یک data access technology . بنابراین باید دقیقا linq to sql یا linq to entities را مشخص کرد.
سابقه نزدیک به یک دهه پروژه اصلی Hibernate که توسط جاوا کارها توسعه داده شده، در این فریم ورک لحاظ شده که از هر لحاظ نسبت به LINQ to entities اون رو پخته‌تر کرده. ضمنا پروایدر LINQ هم برای NH اخیرا توسعه داده شده و از این لحاظ کم و کسری ندارد.
linq to sql برای اس کیوال سرور توسعه داده شد. بعد مایکروسافت اومد اون رو با linq to entities تکمیل کرد (البته linq to sql مطابق وبلاگ رسمی برنامه نویس‌های MS هنوز هم توسعه پیدا می‌کنه و در دات 4 شاهد اون خواهیم بود) و توسط linq to entities امکان استفاده از سایر دیتابیس‌ها هم فراهم شده البته اگر پروایدر آن موجود باشد که تعدادی از آن‌ها هم تجاری هستند. اما با NH این مشکل رو ندارید چون تقریبا همه نوع دیتابیس معروفی را ساپورت می‌کند و رایگان هم هست.
learning curve مربوط به NH بیشتر است از سایر orm ها.
NH از دات نت فریم 2 به بعد را پشتیبانی می‌کند اما linq to entities فقط از دات نت فریم ورک سه و نیم سرویس پک یک به بعد به صورت کامل در دسترس است.

در کل در گوگل linq vs nhibernate را جستجو کنید.

@LoveAjax
محیط مدیوم تراست، امکان ریفلکشن رو حذف می‌کنه و این مورد برای NH و تمام ORM های دیگر نیز مساله ساز خواهد بود. اما برای NH راه حل دارد مطابق مستندات آن:
http://nhforge.org/wikis/howtonh/run-in-medium-trust.aspx

نویسنده: محمد امین شریفی
تاریخ: ۱۳۸۸/۰۷/۱۸ ۱۹:۲۶:۵۱

درباره entity های ماکروسافت هم اگر امکانش هست بنویسید.
فناوری های LINQ to entity و ADO.net entity
برای کسی که فقط با MSSQL کار میکند،آیا فناوری های بالا کمبودی نسبت به NHibernate دارند؟
منظور از هاست های medium trust چیست،یعنی ORM ها را نمی توان روی آنها اجرا کرد؟

@};-

نویسنده: Alex
تاریخ: ۱۳۸۹/۰۱/۱۶ ۱۶:۴۸:۰۰

آقای نصیری میتونید مزایای Fluent رو نسبت به خود NHibernate رو بگید یا مرجعی معرفی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۱/۱۶ ۲۰:۲۱:۰۰

در NHibernate سنتی کار ساخت نگاشت‌ها توسط یک سری فایل xml صورت می‌گیرد که ممکن است حین تهیه اولیه پر از اشتباهات تایپی و غیره باشند.این نوع فایل‌ها تحت کنترل کامپایلر نبوده و در حین کار مشکلات آن‌ها مشخص می‌شود.
در Fluent NHibernate کار تعریف نگاشت‌ها با استفاده از کدهای strongly typed دات نتی صورت می‌گیرد که بلافاصله تحت کنترل کامپایلر هستند. همچنین مبحث Auto Mapping آن را می‌توانید در قسمت‌های بعد مطالعه کنید. امکان unit test نوشتن برای نگاشت‌های این روش بدون حتی درج یک رکورد در دیتابیس میسر است که باز هم در طی چند قسمت به آن پرداخته شده.
با توجه به اینکه در روش دوم تعریف نگاشت‌ها، بلافاصله تحت نظر کامپایلر است امکان refactoring ساده‌تر آن نیز مهیا است.
در روش Fluent اگر علاقمند بودید که این فایل‌های XML را هم مشاهده کنید به قسمت Mappings در Fluently.Configure خود، متد ExportTo را اضافه کنید.

نویسنده: Alex
تاریخ: ۱۳۸۹/۰۱/۱۷ ۰۷:۲۵:۱۷

بینهایت ممنون از توضیحاتی که دادید.

نویسنده: peyman naji
تاریخ: ۱۳۸۹/۰۷/۰۶ ۱۱:۵۰:۳۲

با سلام

در ورژن آلفا 3.0 دیگه خبری از FluentNHibernate.dll نیست چکار باید کرد مهندس ؟ کلا قسمت اول رو نتونستم

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۰۶ ۱۲:۳۵:۲۵

سلام،
سورس هر دو را دریافت کنید. سپس FluentNHibernate را بر اساس نگارش 3 مجددا کامپایل کرده و استفاده کنید :
+

نویسنده: fateme
تاریخ: ۱۳۸۹/۰۹/۲۱ ۱۱:۵۱:۴۱

جناب آقای نصیری
با سلام
من نمیتونم پروژه رو بگیرم وقتی آدرسو در checkout وارد میکنم در قسمت setting هم تنظیماتو انجام میدم error عدم دسترسی به آدرس رو میده چکار کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۲۱ ۱۳:۰۳:۲۷

آدرس جدید: (+)

نویسنده: fateme
تاریخ: ۱۳۸۹/۰۹/۲۲ ۱۱:۴۵:۲۶

با سلام
من آدرس جدیدی که دادید رو در checkout وارد که می کنم error زیر رو بهم میده
error validating server certificate for http://github.com:443:
unknown certificate issuer
واقعا ممنونم که جواب سوالات رو میدید

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۲۲ ۱۲:۳۳:۰۳

سلام، علت اینکه از گوگل کد نمی‌تونید فایلی دریافت کنید این است که گوگل ما رو خیلی وقت است تحریم کرده و درب گوگل کد به روی ایرانی‌ها بسته است. به همین جهت عرض کردم که نیاز به پروکسی دارید و نحوه‌ی ورود اطلاعات پروکسی به TortoiseSVN را نیز ذکر کردم.
در مورد GitHub (آدرس جدید) با استفاده از مرورگر وب به آن وارد شوید. بالای صفحه یک دکمه‌ی دریافت هست. به این صورت به سادگی کل مجموعه رو به شکل یک فایل zip می‌تونید دریافت کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۲۴ ۱۲:۱۴:۵۶

pre-release binaries (v1.2) with NH3 support