عنوان: **NHibernate و مدیریت خودکار تغییرات ساختار بانک اطلاعاتی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۱/۳۰ ۱۱:۰۶:۰۰
آدرس: www.dotnettips.info
گروه‌ها: NHibernate

یکی از دردهای عظمایی که حین کار با بانک‌های اطلاعاتی رابطه‌ای وجود دارد، هماهنگ نبودن دیتابیس توسعه، با دیتابیس کاری است. البته ابزارهای متعددی برای تهیه Diff بین این دو وجود دارند. ولی زمانیکه قرار باشد این کار را در چندجا هم انجام دهیم، باز هم مشکل خواهد بود.
با NHibernate می‌شود کل این مساله را فراموش کرد! می‌شود راحت خاصیتی را به کلاسی اضافه کرد و در اولین بار اجرای برنامه، خود NHibernate هماهنگ سازی‌ها را انجام دهد. فیلد اضافه کند. جدول اضافه کند. روابط مرتبط را اضافه کند. یعنی تا این حد که ما فقط فایل اجرایی برنامه را به روز کنیم، کافی باشد. البته در لابلای مطالبی که تا به حال در مورد NHibernate در این سایت منتشر شده به این موضوع هم پرداخته شده و مطلب جاری، خلاصه‌ی بزرگنمایی شده آن‌ها است.

اولین قدم: آیا ساختار دیتابیس جاری، با مدل برنامه تطابق دارد؟
قبل از اینکه از NHibernate بخواهیم ساختار بانک اطلاعاتی ما را تغییر دهید، باید بدانیم که آیا واقعا نیازی به اینکار هست یا خیر؟ توضیحات بیشتر در مورد روش تشخیص در اینجا: (^)

قدم دوم: اگر ساختار دیتابیس جاری با مدل برنامه تطابق ندارد، چگونه باید آن‌را به صورت خودکار به روز کرد؟
متد زیر بر اساس Configuration ابتدایی بانک اطلاعاتی و نگاشت‌های شما، کار به روز رسانی خودکار ساختار بانک اطلاعاتی را انجام خواهد داد:

```
public void UpdateDatabaseSchema(NHibernate.Cfg.Configuration config)
{
     var schemaUpdate = new NHibernate.Tool.hbm2ddl.SchemaUpdate(config);
     schemaUpdate.Execute(script: false, doUpdate: true);
}
```

یک نکته را هم باید درنظر داشت. در این روش هیچ فیلد و جدولی حذف نمی‌شود و به این ترتیب، جهت امنیت بیشتر طراحی شده. اگر واقعا نیاز داشتید فیلد یا جدولی را حذف کنید باید دستی، همانند سابق اقدام کنید.

قدم سوم: چگونه و در کجا، دو قدم قبل را با برنامه یکپارچه کنیم؟
بلافاصله پس از ایجاد SessionFactory در برنامه، متد زیر را فراخوانی کنید:

```
public void TryValidateAndUpdateDatabaseSchema(NHibernate.Cfg.Configuration config)
{
     try
     {
          ValidateDatabaseSchemaAgainstMappings();
     }
     catch
     {
          UpdateDatabaseSchema(config);
     }
}
```

متد ValidateDatabaseSchemaAgainstMappings در صورت عدم تطابق مدلی با بانک اطلاعاتی، یک exception را صادر می‌کند. بنابراین در اینجا کافی است متد UpdateDatabaseSchema را در قسمت catch فراخوانی کرد.

و از این پس دیگر می‌توانید به روز رسانی ساختار بانک اطلاعاتی برنامه را فراموش کنید! فیلد اضافه کنید، کلاس اضافه کنید، تمام این‌ها در اولین بار اجرای برنامه به روز شده، به صورت خودکار به بانک اطلاعاتی اعمال خواهند شد.

نظرات خوانندگان

نویسنده: MehdiPayervand
تاریخ: ۰۹:۴۶:۵۳ ۱۳۹۰/۱۲/۰۲

در بیشتر شرکت هایی که به طریقی میخوان از بانک هم ورژن داشته باشند معمولا کنار هر پروژه یک پروژه از نوع بانک ساخته میشه (فقط MS SQL) ولی با این امکان NHibernate دیگه احتیاجی به نگهداری یک پروژه موازی نیست، همه تسک ها بصورت مجتمع و مدیریت شده در کلاسهای سطح سلوشن نگهداری میشه.
واقعا پست هاتون برای توسعه دهنده ها یه نوع واکسیناسیون میمونه ;)

نویسنده: رامین علیرضایی
تاریخ: ۱۹:۹ ۱۳۹۴/۰۶/۲۸

ممنون از مطلب خوبتان.
جهت تکمیل مطلب، برای کنترل نسخه تغییرات بانک اطلاعاتی میشه از Fluent Migrator استفاده کرد.