

پیشتر مطلبی را در مورد 18 مقاله‌ای که اکثر حالت‌های Mapping موجود در NHibernate را خلاصه کرده بود، [مطالعه کردید](#). یک مورد هم در این مطلب به نظر در مقایسه با Fluent NHibernate در نظر گرفته نشده است و آن هم بحث [AutoMapping](#) است. Fluent NHibernate این قابلیت را دارد که بر اساس تعاریف کلاس‌های شما و روابط بین آن‌ها به صورت خودکار نگاشت‌ها را تشکیل دهد. یعنی خودش مباحث ارتباط‌های یک به چند و چند به چند و غیره را در پشت صحنه به صورت خودکار تولید کند؛ بدون حتی یک سطر کدنویسی اضافی. فقط حداکثر یک سری [IAutoMappingOverride](#) و همچنین تعدادی [Convention](#) نامگذاری را هم می‌توان جهت تنظیمات این سیستم تمام خودکار اعمال کرد. مثلاً توسط [IAutoMappingOverride](#)، یکی از خاصیت‌های کلاس را به صورت Unique معرفی کرد و مابقی هم به صورت خودکار توسط قابلیت Automapping نگاشت می‌شوند. یا توسط Convention نامگذاری سفارشی خود، به Fluent NHibernate اعلام می‌کنیم که من علاقمندم نام مثلاً کلیدهای خارجی تشکیل شده بجای اعدادی منحصر بفرد، از روش ویژه‌ای که تعیین می‌کنم، ساخته شوند. اینجا است که به نظر من کار با NHibernate حتی از Entity framework هم ساده‌تر است (یا ساده‌تر شده است).

قابلیت AutoMapping یاد شده، در سیستم جدید توکار Mapping by code هم وجود دارد. فقط چون جایی به صورت درست و درمان مستند نشده، همه دور خودشان می‌چرخند! به همین جهت مثالی را در این زمینه آماده کردم که موارد زیر را پوشش می‌دهد:

- نحوه اعمال تنظیمات بانک اطلاعاتی با کدنویسی در NH3,2
- نحوه یکپارچه سازی این روش جدید با کتابخانه [NHibernate Validator](#)
- استفاده از NHibernate Validator جهت تنظیم طول فیلدهای بانک اطلاعاتی تولیدی به صورت خودکار
- نحوه تعریف قراردادهای نامگذاری ویژه (مثلاً نام جداول تولید شده به صورت خودکار، برخلاف نام موجودیت‌ها، جمع باشد نه مفرد)
- اضافه کردن قابلیت تشخیص many-to-many به auto-mapping موجود در NH3,2 (برای تشخیص این مورد به کمک امکانات مهبیای پیش فرض NH3,2، باید اندکی کدنویسی کرد)
- نحوه بازنویسی قراردادهای پیش فرض auto-mapping. مثلاً اگر قرار باشد همه چیز خودکار شود اما یکی از فیلدها به صورت unique معرفی شود چکار باید کرد.
- [نحوه](#) ذخیره اطلاعات mapping کامپایل شده در فایل فایل و سپس بارگذاری خودکار آن در حین اجرای برنامه جهت بالا بردن سرعت بارگذاری اولیه برنامه.

[NHibernate 3.2 Auto-Mapping Sample](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۱۰ ۲۲:۴۰:۰۱

این مثال رو مجدداً به روز کردم. این قابلیت‌ها به آن اضافه شده:
- استفاده از یک کلاس پایه جهت قرار دادن یک سری خواص تکراری در آن، بدون لحاظ شدن این کلاس پایه به عنوان یک موجودیت مستقل.
- ارث بری از یک کلاس پایه و نگاشت خودکار آن.
- پشتیبانی از موجودیت‌های خود ارجاعی (سلسله مراتبی).

نویسنده: MehdiPayervand
تاریخ: ۱۳۹۰/۱۲/۱۱ ۱۲:۴۷:۴۰

ازتون بابت اشتراک دانشتون ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۱۲ ۱۰:۳۹:۴۵

EnumConvention رو هم به مثال اضافه کردم. به صورت پیش فرض Enumها در NH به اعداد Map می‌شوند. اگر نیاز بود به رشته‌های معادل مپ شوند، این EnumConvention خودکار مفید خواهد بود.

نویسنده: MehdiPayervand
تاریخ: ۱۳۹۰/۱۲/۱۲ ۱۲:۲۸:۳۷

اتفاقاً دیروز بدنال نمونه مناسب از استفاده مپ برای Enum می‌گشتم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۱۶ ۰۹:۱۸:۴۸

مثال رو مجدداً به روز کردم. Component mapping و همچنین یک روال نامگذاری سفارشی خودکار هم برای آن اضافه شد.
Component mapping مربوط به حالتی است که شما یک سری خواص را در یک کلاس تعریف می‌کنید اما قصد ندارید این‌ها واقعاً تبدیل به یک جدول مجزا در بانک اطلاعاتی شوند. می‌خواهید این خواص دقیقاً در همان جدول اصلی کنار مابقی خواص قرار گیرد. اما در طرف کدهای ما به شکل یک کلاس مجزا تعریف شود.

نویسنده: علی رضا71
تاریخ: ۱۳۹۲/۱۱/۰۶ ۲۳:۰۹

چرا به این بخش رسیدگی نمی‌کنید؟ آیا این روش منسوخ شده؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۲۳:۱۱

نه. [زمانه تغییر کرده](#).