عنوان: **استفاده از فیلدهای XML در NHibernate**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۳/۳۰ ۱۳:۳۳:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: NHibernate

در مورد طراحی یک برنامه "فرم ساز" در مطلب قبلی بحث شد ... حدودا سه سال قبل اینکار را برای شرکتی انجام دادم. یک برنامه درخواست خدمات نوشته شده با ASP.NET که مدیران برنامه می‌توانستند برای آن فرم طراحی کنند؛ فرم درخواست پرینت، درخواست نصب نرم افزار، درخواست وام، درخواست پیک، درخواست آژانس و ... فرم‌هایی که تمامی نداشتند! آن زمان برای حل این مساله از فیلدهای XML استفاده کردم.
فیلدهای XML قابلیت نه چندان جدیدی هستند که از SQL Server 2005 به بعد اضافه شده‌اند. مهم‌ترین مزیت آن‌ها هم امکان ذخیره سازی اطلاعات هر نوع شیءایی به عنوان یک فیلد XML است. یعنی همان زیرساختی که برای ایجاد یک برنامه فرم ساز نیاز است. ذخیره سازی آن هم آداب خاصی را طلب نمی‌کند. به ازای هر فیلد مورد نظر کاربر، یک نود جدید به صورت رشته معمولی باید اضافه شود و نهایتا رشته تولیدی باید ذخیره گردد. از دید ما یک رشته است، از دید SQL Server یک نوع XML واقعی؛ به همراه این مزیت مهم که به سادگی می‌توان با T-SQL/XQuery/XPath از جزئیات اطلاعات این نوع فیلدها کوئری گرفت و سرعت کار هم واقعا بالا است؛ به علاوه بر خلاف مطلب قبلی در مورد dynamic components ، اینبار نیازی نیست تا به ازای هر یک فیلد درخواستی کاربر، واقعا یک فیلد جدید را به جدول خاصی اضافه کرد. داخل این فیلد XML هر نوع ساختار دلخواهی را می‌توان ذخیره کرد. به عبارتی به کمک فیلدهایی از نوع XML می‌توان داخل یک سیستم بانک اطلاعاتی رابطه‌ای، schema-less کار کرد (un-typed XML) و همچنین از این اطلاعات ویژه، کوئری‌های پیچیده هم گرفت.
تا جایی که اطلاع دارم، چند شرکت دیگر هم در ایران دقیقا از همین ایده فیلدهای XML برای ساخت برنامه فرم ساز استفاده کرده‌اند ...؛ البته مطلب جدیدی هم نیست؛ برنامه‌های فرم ساز اوراکل و IBM هم سال‌ها است که از XML برای همین منظور استفاده می‌کنند. مایکروسافت هم به همین دلیل (شاید بتوان گفت مهم‌ترین دلیل وجودی فیلدهای XML در SQL Server)، پشتیبانی توکاری از XML به عمل آورده است.
یا روش دیگری را که برای طراحی سیستم‌های فرم ساز پیشنهاد می‌کنند استفاده از بانک‌های اطلاعاتی مبتنی بر key-value مانند Redis یا RavenDb است؛ یا استفاده از بانک‌های اطلاعاتی schema-less واقعی مانند CouchDb .

خوب ... اکنون سؤال این است که NHibernate برای کار با فیلدهای XML چه تمهیداتی را درنظر گرفته است؟
برای این منظور خاصیتی را که قرار است به یک فیلد از نوع XML نگاشت شود، با نوع XDocument مشخص خواهیم ساخت:

```
using System.Xml.Linq;

namespace TestModel
{
   public class DynamicTable
   {
       public virtual int Id { get; set; }
       public virtual XDocument Document { get; set; }
   }
}
```

سپس باید جهت معرفی این نوع ویژه، به صورت صریح از XDocType استفاده کرد؛ یعنی نکته‌ی اصلی، استفاده از CustomType مرتبط است:

```
using FluentNHibernate.Automapping;
using FluentNHibernate.Automapping.Alterations;
using NHibernate.Type;

namespace TestModel
{
   public class DynamicTableMapping : IAutoMappingOverride<DynamicTable>
   {
       public void Override(AutoMapping<DynamicTable> mapping)
       {
```

```
            mapping.Id(x => x.Id);
            mapping.Map(x => x.Document).CustomType<XDocType>();
        }
    }
}
```

البته لازم به ذکر است که دو نوع NHibernate.Type.XDocType و NHibernate.Type.XmlDocType برای کار با فیلدهای XML در NHibernate وجود دارند. XDocType برای کار با نوع System.Xml.Linq.XDocument طراحی شده است و XmlDocType مخصوص نگاشت نوع System.Xml.XmlDocument است.

اکنون اگر به کمک کلاس SchemaExport ، اسکریپت تولید جدول متناظر با اطلاعات فوق را ایجاد کنیم به حاصل زیر خواهیم رسید:

```
if exists (select * from dbo.sysobjects
                where id = object_id(N'[DynamicTable]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
        drop table [DynamicTable]

    create table [DynamicTable] (
        Id INT IDENTITY NOT NULL,
       Document XML null,
       primary key (Id)
    )
```

یک سری اعمال متداول ذخیره سازی اطلاعات و تهیه کوئری نیز در ادامه ذکر شده‌اند:

```
//insert
object savedId = 0;
using (var session = sessionFactory.OpenSession())
{
    using (var tx = session.BeginTransaction())
    {
         var obj = new DynamicTable
         {
             Document = System.Xml.Linq.XDocument.Parse(
                    @"<Doc><Node1>Text1</Node1><Node2>Text2</Node2></Doc>"
                    )
         };
         savedId = session.Save(obj);
         tx.Commit();
     }
}

//simple query
using (var session = sessionFactory.OpenSession())
{
    using (var tx = session.BeginTransaction())
    {
         var entity = session.Get<DynamicTable>(savedId);
         if (entity != null)
         {
            Console.WriteLine(entity.Document.Root.ToString());
         }

         tx.Commit();
     }
}

//advanced query
using (var session = sessionFactory.OpenSession())
{
    using (var tx = session.BeginTransaction())
    {
        var list = session.CreateSQLQuery("select [Document].value('(//Doc/Node1)[1]','nvarchar(255)')
from [DynamicTable] where id=:p0")
                                .SetParameter("p0", savedId)
                                .List();

         if (list != null)
```

```
        {
            Console.WriteLine(list[0]);
        }

        tx.Commit();
    }
}
```

و در پایان بدیهی است که جهت کار با امکانات پیشرفته‌تر موجود در SQL Server در مورد فیلدهای XML (برای نمونه: + و +) باید مثلا رویه ذخیره شده تهیه کرد (یا مستقیما از متد CreateSQLQuery همانند مثال فوق کمک گرفت) و آن‌را در NHibernate مورد استفاده قرار داد. البته به این صورت کار شما محدود به SQL Server خواهد شد و باید در نظر داشت که در کل تعداد کمی بانک اطلاعاتی وجود دارند که نوع‌های XML را به صورت توکار پشتیبانی می‌کنند.

نظرات خوانندگان

نویسنده: Afshar Mohebbi
تاریخ: ۱۳۹۰/۰۳/۳۰ ۲۱:۲۳:۲۷

جالب بود. خصوصاً این که در مورد «فرم ساز»ها صحبت می‌کرد.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۰/۰۳/۳۱ ۱۰:۴۰:۴۱

سلام.
من nHibernate بلد نیستم اما قسمتی که درباره فرم ساز ها صحبت کردید جالب بود.
این فرم سازی که شما ازش صحبت کردید User Mode بود حالا اگر فرم سازی بخواهیم توسعه بدیم که Developer Mode باشه نظرتون چیه؟
مثلا از روی یک Table فرم خام رو صورت Html در بیاورد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۳/۳۱ ۱۰:۴۵:۱۳

به ASP.NET MVC کوچ کنید. از روی مدل شما فرم‌های insert/delete/update به همراه اعتبار سنجی و غیره رو همه رو یکجا با ابزارهای توکاری که داره تولید می‌کنه. حتی مقادیر وارد شده توسط کاربر رو هم به صورت خودکار به فیلدهای مدل انتساب می‌ده (model binding).