عنوان: **پراپرتی سفارشی در EF Database First**
نویسنده: جمشیدی فر
تاریخ: ۱۳۹۲/۰۸/۰۵ ۱۸:۲۰
آدرس: www.dotnettips.info
برچسب‌ها: Entity framework, LINQ to Entity, EF Database First

فرض کنید در روش EF Database First می‌خواهید فیلدی به مدل اضافه شود که در دیتابیس وجود ندارد، درواقع فیلدی محاسباتی به مدل اضافه کنید.راه حل چیست؟ اولین روشی که ممکن است به ذهن برسد این است که به کلاس مدل که از جدول دیتابیس ساخته شده، فیلدی محاسباتی اضافه می‌کنیم.

```
public class Person
{
    public string FullName {
        get
        {
            return FirstName + " " + LastName;
        }
    }
}
```

به نظر راه حل درستی می‌رسد، اما مشکل این روش چیست؟ اگر مدل برنامه از روی دیتابیس بروزشود، چه اتفاقی می‌افتد؟ خب قاعدتا این فیلد محاسباتی از دست می‌رود و باید دوباره آن را به کلاس مدل جدید و بروز شده اضافه کنیم. که به نظر راه حل منطقی و خوبی نمیرسد. در چنین مواقعی می‌توان به جای اینکه این پراپرتی را به کلاس تولید شده از روی دیتابیس اضافه کرد، این فیلد را به یک Partial Class از کلاس تولید شده که درهمان پروژه و فضای نام کلاس تولید شده از دیتابیس قراردارد، اضافه کرد. به این ترتیب با هر بار بروز شدن مدل از روی دیتابیس، این فیلد از بین نمی‌رود.

```
public partial class Person
{
    public string FirstName {get; set;}
    public string LastName {get; set; }
}
public partial class Person
{
    public string FullName {
        get
        {
            return FirstName + " " + LastName;
        }
    }
}
```

اما این روش محدودیت هایی نیز دارد :

1. همه قسمت‌های Partial Class باید در یک اسمبلی باشند.
2. پراپرتی‌های درون Partial Class در دیتابیس ذخیره نمی‌شوند.
3. ونیز این پراپرتی‌ها در عبارات Linq قابل استفاده نیستند.چون عبارت Linq در نهایت به یک رشته SQL تبدیل شده و در دیتابیس اجرا می‌شود.البته با فرض این که دیتاپرووایدر، SQL باشد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۸/۰۵ ۲۱:۴۱

در مورد روش دوم؛ (با توجه به جمله بندی) آیا در روش اول این خاصیت‌های محاسباتی در بانک اطلاعاتی ذخیره می‌شوند؟ ضمنا جهت تکمیل بحث، این خاصیت‌ها (هر دو حالت) در عبارات LINQ to Objects قابل استفاده هستند.

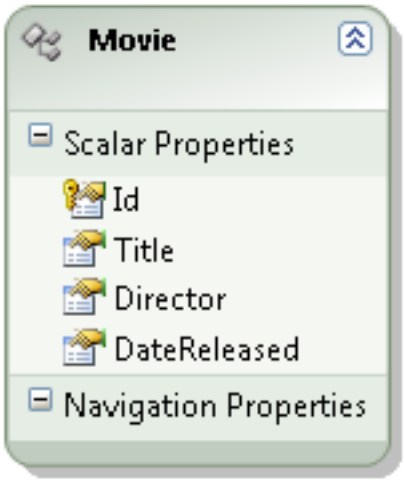
نویسنده: جمشیدی فر
تاریخ: ۱۳۹۲/۰۸/۰۶ ۸:۳۸

در مورد سوال اول، خیر؛ در روش اول هم این خاصیت‌ها در دیتابیس ذخیره نمی‌شوند.اینجا مساله این است که ما می‌خواهیم با هر بار اپدیت مدل از دیتابیس، فیلد محاسباتی دوباره محاسبه شود و ازبین نرود.

درمورد سوال دوم، شما درست می‌فرمایید. من باید Linq رو به Linq to Entity تغییر بدم.

عنوان: **استفاده از DataAnnotations در EF DataBase First**
نویسنده: حسین حسینی 128
تاریخ: ۱۳۹۲/۰۸/۲۸ ۱۲:۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, ASP.Net MVC, EF Database First

همانطور که میدانیم DataAnnotations برای فیلدهای مدل‌ها در MVC وقتی که از EF Code First استفاده کنیم کار ما را برای اعتبارسنجی بسیار ساده میکنند. اما وضع در EF Database First متفاوت است زیرا اگر مدلی را که برنامه برایمان میسازد را به روز کنیم (توسط Updata Model) تمام اعتبارسنجی هایی که نوشته بودیم پاک شده و مدل خام دوباره برای ما تولید میشود . برای رفع این مشل باید از PartialClass استفاده نماییم تا بتوانیم همیشه اعتبارسنجی‌ها را داشته باشیم :




```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

namespace MvcApplication1.Models
{
    [MetadataType(typeof(MovieMetaData))]
    public partial class Movie
    {
    }


    public class MovieMetaData
    {
        [Required]
        public object Title { get; set; }

        [Required]
        [StringLength(5)]
        public object Director { get; set; }


        [DisplayName("Date Released")]
        [Required]
        public object DateReleased { get; set; }
    }

}
```

لینک مطلب اصلی : http://www.asp.net/mvc/tutorials/older-versions/models-(data)/validation-with-the-data-annotation-validators-cs

نظرات خوانندگان

نویسنده: لیلا
تاریخ: ۱۳۹۲/۰۸/۲۹ ۱۹:۲۳

با تشکر
اگر دیتابیس تغییر کند باید PartialClass تغییر را دستی تغییر بدهیم؟ اگر پاسخ بله می‌باشد راهی وجود دارد برای اعمال تغییر ات به صورت خودکار؟
ایا توصیه می‌شود از view model به جای PartialClass برای اعتبارسنجی استفاده شود؟

نویسنده: حسین حسینی 128
تاریخ: ۱۳۹۲/۰۸/۲۹ ۲۳:۹

بله باید دستی تغییر کند ، راه خودکاری نمیدانم متاسفانه
کلا بهتر است از viewmodel به جای استفاده از خود مدل استفاده کرد ولی به هرحال viewmodel نیز پس از تغییر دیتابیس باید دستی تغییر کند

نویسنده: bahman
تاریخ: ۱۳۹۲/۰۹/۱۲ ۱۸:۲۳

سلام.
به خاطر مطلب خوبتون تشکر میکنم.
من یه پروژه 2 لایه دارم که مدل در لایه DLL قرار گرفته.
و یک لایه که پروژه MVC داخل اون قرار گافته.
میخواستم بدنم که متادیتا باید داخل کدوم لایه نوشته بشه؟

نویسنده: حسین حسینی 128
تاریخ: ۱۳۹۲/۱۰/۱۷ ۱۷:۵۹

متادیتا داخل لایه DLL باشه بهتره