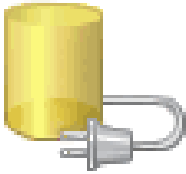


عنوان: سرویس جمع و مفرد سازی اسامی  
نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۱۰/۲۸ ۱۰:۲۰:۰۰  
آدرس: [www.dotnettips.info](http://www.dotnettips.info)  
برچسب‌ها: NHibernate

اگر به Entity data model wizard در VS.Net 2010 دقت کرده باشید، گزینه‌ی "Pluralize or singularize generated object names" نیز به آن اضافه شده است:

# Entity Data Model Wizard



## Choose Your Database Objects

Which database objects do you want to include in your model?

☐

Tables

☐

Views

☐

Stored Procedures

☒

Pluralize or singularize generated object names

Model Namespace:

NorthwindModel

< Previous

Next >

این مورد از این جهت حائز اهمیت است که عموماً نام جداول در بانک اطلاعاتی، جمع است و نام کلاس متناظر ایجاد شده برای آن در کدهای برنامه بهتر است مفرد باشد. برای مثال نام جدول، Customers است و نام کلاس آن بهتر است Customer تعریف گردد. به این صورت کار کردن با آن توسط یک ORM با معناتر خواهد بود؛ زیرا زمانیکه یک وهله از شیء Customer ایجاد می‌شود، فقط یک رکورد از بانک اطلاعاتی مد نظر است؛ در حالیکه یک جدول مجموعه‌ای است از رکوردها.

زبان انگلیسی هم پر است از اسامی جمع و مفرد باقاعده و بی‌قاعده و کل عملیات با اضافه و حذف کردن یک s و یا es پایان نمی‌یابد؛ برای مثال phenomenon و phenomena را در نظر بگیرد تا Money و Moneys.

این امکان مهیا شده توسط Entity Framework 4.0 یا همان EF v2 با برنامه نویسی هم [قابل دسترسی است](#) و در اسمبلی System.Data.Entity.Design.dll و فضای نام System.Data.Entity.Design.PluralizationServices قرار گرفته است.

این اسمبلی جزئی از دات نت 4 است و اگر آن را توسط گزینه‌ی Add references در VS.NET مشاهده نمی‌کنید، علت آن است که در تنظیمات پروژه جاری، گزینه‌ی Target framework بر روی Client profile قرار گرفته است که باید به دات نت 4 کامل تغییر یابد.

استفاده از آن هم به صورت زیر است:

```
using System;
using System.Data.Entity.Design.PluralizationServices;
using System.Globalization;

namespace PluralizationServicesTest
{
    class Program
    {
        static void Main(string[] args)
        {
            var service = PluralizationService.CreateService(CultureInfo.GetCultureInfo("en"));
            Console.WriteLine(service.Pluralize("mouse"));
            Console.WriteLine(service.IsPlural("phenomena"));
        }
    }
}
```

ملاحظات:

این روش فعلاً به زبان انگلیسی محدود است و اگر Culture را به مورد دیگری تنظیم کنید با خطای "We don't support locales other than English yet" متوقف خواهید شد.

روش دیگر:

کتابخانه‌ی سورس باز Castle ActiveRecord نیز دارای کلاسی است به نام Inflector که برای همین منظور طراحی شده است:

[Inflector.cs](#)

کاربرد آن در Fluent NHibernate

در Fluent NHibernate کار نگاشت کلاس‌ها به جداول به صورت خودکار صورت می‌گیرد و همچنین تولید ساختار بانک اطلاعاتی نیز به همین نحو می‌باشد. اما می‌توان تولید نام جداول را سفارشی نیز نمود. برای مثال از کلاس Book به صورت خودکار ساختار جدولی به نام Books را تولید کند:

```
using FluentNHibernate.Conventions;
using FluentNHibernate.Conventions.Instances;
using NHibernate.Helper.Toolkit;

namespace NHibernate.Helper.MappingConventions
{
    public class TableNameConvention : IClassConvention
    {
        public void Apply(IClassInstance instance)
        {
            instance.Table(Inflector.Pluralize(instance.EntityType.Name));
        }
    }
}
```

و برای تزریق آن خواهیم داشت:

```
... = new AutoPersistenceModel()  
    .Where(...)  
    .Conventions.Setup(c =>c.Add<TableNameConvention>())  
    .AddEntityAssembly(...)  
...
```