عنوان: **اجرای Stored Procedure با چند نوع مقدار برگشتی توسط EF CodeFirst**
نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۷/۲۵ ۱۶:۲۹
آدرس: www.dotnettips.info
گروه‌ها: C#, Entity framework, EF Code First, SQL Server

فرض کنید Stored Procedure ی با چند مقدار برگشتی را می‌خواهیم در EF CodeFirst مورد استفاده قرار دهیم. برای مثال Stored Procedure زیر را در نظر بگیرید:

```
CREATE PROCEDURE [dbo].[GetAllBlogsAndPosts]
AS
    SELECT * FROM dbo.Blogs
    SELECT * FROM dbo.Posts
```

Stord Procedure ی که توسط این دستور ساخته می‌شود تمام رکوردهای جدول Blogs و تمامی رکوردهای جدول Posts را واکشی کرده و به عنوان خروجی برمیگرداند (دو خروجی متفاوت). روش فراخوانی و استفاده از داده‌های این StoredProcedure در EF CodeFirst به صورت زیر است :
تعریف دو کلاس مدل Blog و Post به ترتیب برای نگهداری اطلاعات وبلاگ‌ها و پست‌ها در زیر آمده است. در ادامه نیز تعریف کلاس BloggingContext را مشاهده می‌کنید.

```
public class Blog
    {
        public int BlogId { get; set; }
        public string Name { get; set; }

        public virtual List<Post> Posts { get; set; }
    }

    public class Post
    {
        public int PostId { get; set; }
        public string Title { get; set; }
        public string Content { get; set; }

        public int BlogId { get; set; }
        public virtual Blog Blog { get; set; }
    }

    public class BloggingContext : DbContext
    {
        public DbSet<Blog> Blogs { get; set; }
        public DbSet<Post> Posts { get; set; }
    }



using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Data.Objects;

namespace Sproc.Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var db = new BloggingContext())
            {
                 db.Database.Initialize(force: false);

                var cmd = db.Database.Connection.CreateCommand();
                cmd.CommandText = "[dbo].[GetAllBlogsAndPosts]";

                try
                {
                    // اجرای پروسیجر
                    db.Database.Connection.Open();
                    var reader = cmd.ExecuteReader();

                    // خواند رکوردهای blogs
```

```
                    var blogs = ((IObjectContextAdapter)db)
                        .ObjectContext
                        .Translate<Blog>(reader, "Blogs", MergeOption.AppendOnly);

                    foreach (var item in blogs)
                    {
                        Console.WriteLine(item.Name);
                    }

                    // پرش به نتایج بعدی (همان Posts)
                    reader.NextResult();
                    var posts = ((IObjectContextAdapter)db)
                        .ObjectContext
                        .Translate<Post>(reader, "Posts", MergeOption.AppendOnly);

                    foreach (var item in posts)
                    {
                        Console.WriteLine(item.Title);
                    }
                }
                finally
                {
                    db.Database.Connection.Close();
                }
            }
        }
    }
```

در کدهای بالا ابتدا یک Connection به بانک اطلاعاتی باز می‌شود:

```
db.Database.Connection.Open();
```

و پس از آن نوبت به اجرای Stored Procedure می‌رسد:

```
var reader = cmd.ExecuteReader();
```

در کد بالا پس از اجرای Stored Procudure نتایج بدست آمده در یک reader ذخیره می‌شود. شئ reader از نوع DBDataReader می‌باشد. پس از اجرای Stored Procedure و دریافت نتایج و ذخیره سازی در شئی reader ، نوبت به جداسازی رکوردها می‌رسد. همانطور که در تعریف Stored procedure مشخص است این Stored Procedure دارای دو نوع خروجی از نوع‌های Blog و Post می‌باشد و این دو نوع باید از هم جدا شوند.برای انجام این کار از متد Translate شئی Context استفاده می‌شود. این متد قابلیت کپی کردن نتایج موجود از یک شئی DBDataReader به یک شئی از نوع مدل را دارد. برای مثال :

```
var blogs = ((IObjectContextAdapter)db)
           .ObjectContext
           .Translate<Blog>(reader, "Blogs", MergeOption.AppendOnly);
```

در کدهای بالا تمامی رکوردهایی از نوع Blog از شئی reader خوانده شده و پس از تبدیل به نوع Blog درون شئی Blogs ذخیره می‌شود.
پس از آن توسط حلقه foreach محتویات Blogs پیمایش شده و مقدار موجود در فیلد Name نمایش داده می‌شود.

```
foreach (var item in blogs)
  {
            Console.WriteLine(item.Name);
  }
```

با توجه به اینکه حاصل اجرای این Stored Procedure دو خروجی متفاوت بوده است ، پس از پیمایش رکوردهای Blogs باید به سراغ نتایج بعدی که همان رکوردهای Post می‌باشد برویم. برای اینکار از متد NextResult شئی reader استفاده می‌شود:

```
reader.NextResult();
```

در ادامه برای خواندن رکوردهایی از نوع Post نیز به همان روشی که برای Blog انجام شد عمل می‌شود. مطالعه بیشتر

نظرات خوانندگان

نویسنده: ashi mashi
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۴:۳۶

لطفا نحوه اجرا کردن پروسیجرها با مقادیر مختلف در ورودی هم کمی توضیح می‌دادید ممنون می‌شدم.

با تشکر از توضیحات خوبتون،

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۴:۵۴

استفاده مستقیم از عبارات SQL در EF Code first و مثال void runSp آن.

نویسنده: بهمن آبادی
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۱۷

منظور من روش مشابه parameterAttributeها در mapping کردن ورودی‌های پروسیجرها در linq می‌باشد.
نه اینکه صرفا از دستورات sql بصورت command استفاده شود.

مانند استفاده از پروسیجرها با چند ورودی و multiResult بودن آن در linq

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۲۷

در مورد نحوه اجرای رویه‌های ذخیره شده با ورودی‌های مختلف پرسیدید، روش اجراش تا EF 5.0 به همین صورتی است که ملاحظه می‌کنید .

قرار است در EF 6.0 این مساله با Fluent API به نحو دیگری پوشش داده شود.

نویسنده: بهمن آبادی
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۴۱

دقیقا منظورم همون لینکه EF 6.0 که گفتین هستش.
Road Map برای انتشار نسخه جدید EF 6.0 وجود داره که قراره نهایتا تا کی انتشار پیدا بکنه ؟

بازم ممنون.

نویسنده: سید مهدی فاطمی
تاریخ: ۱۳۹۲/۰۷/۰۴ ۲۱:۴۷

تشکر دوست عزیز

اما در مدل من که از دیتابیس گرفتم شی رویه ذخیره شده هم به عنوان یک تابع شبیه سازی شده و مثل شما عمل نکردم اما نمی‌تونم مقادیر رو از این شی بگیرم.

```
var sp1=_db.splogin(user,pas,value1,value2);
```

در توضیح این کد هم بگم که این اس پی 4 پارامتر داره و قراره دو مقدار رو برای من برگردونه .

نویسنده: سانای رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱۹:۳۹

سلام
اگر SP ما بدون پارامتر out باشد و ما در آخر دستورات آن از دستور SCOPE_IDENTITY() استفاده کرده باشیم چگونه می‌توانیم مقدار آن را به دست بیاوریم؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۴/۲۳ ۱۱:۰

در مطلب استفاده مستقیم از عبارات SQL در EF Code first بحث شده. از متد جنریک db.Database.SqlQuery باید استفاده کنید.