عنوان: **نمایش خروجی SQL کدهای Entity framework 6 در کنسول دیباگ ویژوال استودیو**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۱۲ ۱۱:۱۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework

تا قبل از EF 6 برای تهیه لاگ SQL تولیدی توسط Entity framework نیاز بود به ابزارهای ثالث متوسل شد. برای مثال از انواع پروفایلرها استفاده کرد (^ و ^ و ^). اما در EF 6 امکان توکاری به نام Command Interception تدارک دیده شده است تا توسط آن بتوان بدون نیاز به ابزارهای جانبی، به درون سیستم EF متصل شد و دستورات تولیدی آن‌را پیش از اجرای بر روی بانک اطلاعاتی دریافت و مثلا لاگ کرد. در ادامه نمونه‌ای از این عملیات را بررسی خواهیم کرد.

تهیه کلاس SimpleInterceptor

برای اتصال به متدهای اجرای دستورات SQL در EF 6 تنها کافی است یک کلاس جدید را از کلاس پایه DbCommandInterceptor مشتق کرده و سپس متدهای کلاس پایه را override کنیم. در این متدها، فراخوانی متدهای کلاس پایه، معادل خواهند بود با اجرای واقعی دستور بر روی بانک اطلاعاتی. به این ترتیب حتی می‌توان مدت زمان انجام عملیات را نیز بدست آورد. در اینجا command.CommandText معادل است با دستور SQL در حال اجرا و همچنین نیاز است تا تمام سطوح تو در توی استثناهای احتمالی رخ داده را نیز بررسی کرد:

```
using System;
using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;
using System.Diagnostics;
using System.Text;

namespace EFCommandInterception
{
    public class SimpleInterceptor : DbCommandInterceptor
    {
        public override void ScalarExecuting(DbCommand command, DbCommandInterceptionContext<object>
interceptionContext)
        {
            var timespan = runCommand(() => base.ScalarExecuting(command, interceptionContext));
            logData(command, interceptionContext.Exception, timespan);
        }

        public override void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<int>
interceptionContext)
        {
            var timespan = runCommand(() => base.NonQueryExecuting(command, interceptionContext));
            logData(command, interceptionContext.Exception, timespan);
        }

        public override void ReaderExecuting(DbCommand command,
DbCommandInterceptionContext<DbDataReader> interceptionContext)
        {
            var timespan = runCommand(() => base.ReaderExecuting(command, interceptionContext));
            logData(command, interceptionContext.Exception, timespan);
        }

        private static Stopwatch runCommand(Action command)
        {
            var timespan = Stopwatch.StartNew();
            command();
            timespan.Stop();
            return timespan;
        }

        private static void logData(DbCommand command, Exception exception, Stopwatch timespan)
        {
            if (exception != null)
            {
                Trace.TraceError(formatException(exception, "Error executing command: {0}",
command.CommandText));
            }
            else
            {
                Trace.TraceInformation(string.Concat("Elapsed time: ", timespan.Elapsed, " Command: ",
command.CommandText));
            }
```

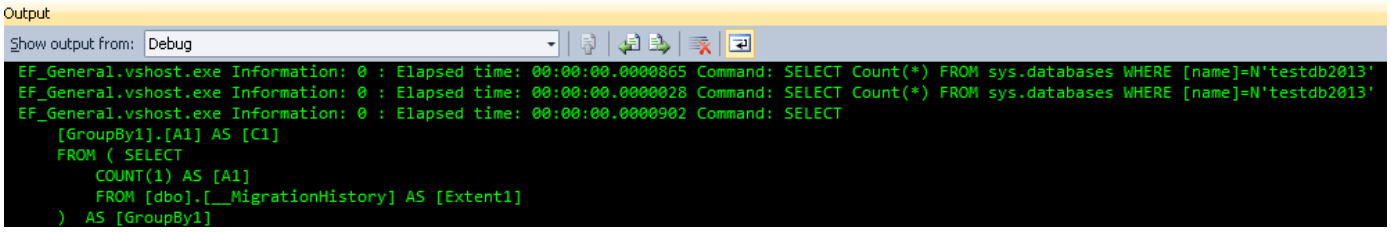
```
        }

        private static string formatException(Exception exception, string fmt, params object[] vars)
        {
            var sb = new StringBuilder();
            sb.Append(string.Format(fmt, vars));
            sb.Append(" Exception: ");
            sb.Append(exception.ToString());
            while (exception.InnerException != null)
            {
                sb.Append(" Inner exception: ");
                sb.Append(exception.InnerException.ToString());
                exception = exception.InnerException;
            }
            return sb.ToString();
        }
    }
}
```

نحوه استفاده از کلاس SimpleInterceptor

کلاس فوق را کافی است تنها یکبار در آغاز برنامه (مثلا در متد Application_Start برنامه‌های وب) به EF 6 معرفی کرد:

```
DbInterception.Add(new SimpleInterceptor());
```

اکنون اگر برنامه را اجرا کنیم، خروجی SQL و زمان‌های اجرای عملیات را در پنجره دیباگ VS.NET می‌توان مشاهده کرد:



نظرات خوانندگان

نویسنده: علیرضا
تاریخ: ۱۳۹۲/۰۸/۱۲ ۱۷:۴۳

سلام، چه کتابی برای EF6 پیشنهاد میکنین؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۱۲ ۱۸:۸

از قسمت اول سری EF Code first شروع کنید. مباحث پایه‌ای همان است. فقط یک سری افزونه بیشتر شده.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۸/۲۱ ۱۰:۳۶

البته با استفاده از خصوصیت Log نیز می‌توانیم دستورات TSQL را در خروجی لاگ کنیم :

```
public MyContext():base("MyConnectionString")
{
            Database.Log = sql => Debug.Write(sql);
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۲:۲۱

یک نکته‌ی تکمیلی

در EF 6.1 به بعد، کل روش ارائه شده در اینجا را می‌توانید به نحو ذیل در فایل کانفیگ برنامه‌های وب یا ویندوزی، برای ذخیره در فایل، فعال کنید (بدون نیاز به کدنویسی اضافه‌تری):

```
<interceptors>
  <interceptor type="System.Data.Entity.Infrastructure.Interception.DatabaseLogger, EntityFramework">
    <parameters>
      <parameter value="C:\Temp\LogOutput.txt"/>
      <parameter value="true" type="System.Boolean"/>
    </parameters>
  </interceptor>
</interceptors>
```