عنوان: **یک دست سازی ی و ک در برنامه‌های Entity framework 6**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۸ ۱۲:۲۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework

تا قبل از EF 6 برای طراحی یک سیستم عمومی تغییر مقادیر ثبت شده در بانک اطلاعاتی، می‌شد با استفاده از امکانات توکار Tracking آن، مقادیر تغییر کرده را یافت و برای مثال ی و ک آن‌ها را پیش از درج در بانک اطلاعاتی، یک دست کرد. در EF 6 با معرفی یک سری interceptor می‌توان به مراحل پیش و پس از اجرای کوئری‌ها دسترسی پیدا کرد. عمده‌ترین کاربرد آن، لاگ کردن SQLهای تولیدی و نوشتن برنامه‌هایی شبیه به EF Profiler است. اما ... استفاده‌ی دیگری را نیز می‌توان از IDbCommandInterceptor جدید آن تدارک دید: دستکاری SQL تولیدی توسط آن پیش از اعمال به بانک اطلاعاتی.

### طراحی یک Interceptor برای یک دست سازی ی و ک

در اینجا کدهای کلاس YeKeInterceptor را ملاحظه می‌کنید. در متدهایی که به کلمه‌ی Executing ختم می‌شوند، می‌توان به دستورات SQL تولید شده توسط EF، پیش از اعمال بر روی بانک اطلاعاتی دسترسی داشت:

```
public class YeKeInterceptor : IDbCommandInterceptor
    {
        public void ReaderExecuting(DbCommand command, DbCommandInterceptionContext<DbDataReader>
interceptionContext)
        {
            command.ApplyCorrectYeKe();
        }

        public void NonQueryExecuted(DbCommand command, DbCommandInterceptionContext<int>
interceptionContext)
        {
        }

        public void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<int>
interceptionContext)
        {
            command.ApplyCorrectYeKe();
        }

        public void ReaderExecuted(DbCommand command, DbCommandInterceptionContext<DbDataReader>
interceptionContext)
        {
        }

        public void ScalarExecuted(DbCommand command, DbCommandInterceptionContext<object>
interceptionContext)
        {
        }

        public void ScalarExecuting(DbCommand command, DbCommandInterceptionContext<object>
interceptionContext)
        {
            command.ApplyCorrectYeKe();
        }
    }
```

DbCommand، حاوی تمام اطلاعاتی است که به آن نیاز داریم؛ شامل CommandText یا همان SQL تولید شده و همچنین command.Parameters برای دسترسی به مقادیر پارامترهای کوئری. نکته‌ی مهم تمام این موارد، قابل ویرایش بودن آن‌ها است.

```
public static class YeKe
    {
        public const char ArabicYeChar = (char)1610;
        public const char PersianYeChar = (char)1740;

        public const char ArabicKeChar = (char)1603;
        public const char PersianKeChar = (char)1705;

        public static string ApplyCorrectYeKe(this object data)
        {
            return data == null ? null : ApplyCorrectYeKe(data.ToString());
        }
```

```
        public static string ApplyCorrectYeKe(this string data)
        {
            return string.IsNullOrWhiteSpace(data) ?
                        string.Empty :
                        data.Replace(ArabicYeChar, PersianYeChar).Replace(ArabicKeChar,
PersianKeChar).Trim();
        }

        public static void ApplyCorrectYeKe(this DbCommand command)
        {
            command.CommandText = command.CommandText.ApplyCorrectYeKe();

            foreach (DbParameter parameter in command.Parameters)
            {
                switch (parameter.DbType)
                {
                    case DbType.AnsiString:
                    case DbType.AnsiStringFixedLength:
                    case DbType.String:
                    case DbType.StringFixedLength:
                    case DbType.Xml:
                        parameter.Value = parameter.Value.ApplyCorrectYeKe();
                        break;
                }
            }
        }
    }
```

در اینجا پیاده سازی متد الحاقی ApplyCorrectYeKe را که در کلاس YeKeInterceptor مورد استفاده قرار گرفت، ملاحظه می‌کنید.
در آن، CommandText و همچنین parameter.Valueها در صورت رشته‌ای بودن، اصلاح می‌شوند.
سربار این روش نسبت به روش‌های پیشین استفاده از Reflection کمتر است. همچنین اشیاء پیچیده و تو در تو را نیز بهتر پشتیبانی می‌کند؛ چون در مرحله Executing، کار پردازش این اشیاء پایان یافته و SQL خام نهایی آن در اختیار ما است.

**نحوه‌ی استفاده از YeKeInterceptor**

در آغاز برنامه، سطر زیر را فراخوانی کنید:

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

**یک مثال کامل برای دریافت**

Sample32.cs

## نظرات خوانندگان

نویسنده: میثم مهربانی
تاریخ: ۱۳۹۳/۰۱/۱۸ ۱۳:۲۹

بر روی SqlServer درست کارمی کند ولی بر روی کانکشن SQL CE پیغام زیر را می‌دهد:

```
System.NotSupportedException was unhandled by user code
  HResult=-2146233067
  Message=DesignTimeVisible
  Source=EntityFramework.SqlServerCompact
  StackTrace:
       at System.Data.Entity.SqlServerCompact.SqlCeMultiCommand.set_CommandText(String value)
       at EfExt.YeKe.ApplyCorrectYeKe(DbCommand command) in e:\test\EfExt\YeKe.cs:line 33
       at EfExt.YeKeInterceptor.ReaderExecuting(DbCommand command, DbCommandInterceptionContext`1
interceptionContext) in e:\MyFilesAndPrograms\WebIO\EfExt\YeKeInterceptor.cs:line 15
       at
System.Data.Entity.Infrastructure.Interception.DbCommandDispatcher.<Reader>b__d(IDbCommandInterceptor
i, DbCommand t, DbCommandInterceptionContext`1 c)
       at
System.Data.Entity.Infrastructure.Interception.InternalDispatcher`1.Dispatch[TTarget,TInterceptionConte
xt,TResult](TTarget target, Func`3 operation, TInterceptionContext interceptionContext, Action`3
executing, Action`3 executed)
       at System.Data.Entity.Infrastructure.Interception.DbCommandDispatcher.Reader(DbCommand command,
DbCommandInterceptionContext interceptionContext)
       at System.Data.Entity.Internal.InterceptableDbCommand.ExecuteDbDataReader(CommandBehavior
behavior)
       at System.Data.Common.DbCommand.ExecuteReader(CommandBehavior behavior)
       at
System.Data.Entity.Core.EntityClient.Internal.EntityCommandDefinition.ExecuteStoreCommands(EntityComman
d entityCommand, CommandBehavior behavior)
  InnerException:
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۸ ۱۴:۳۲

برای SQL CE سطر زیر را حذف کنید:

```
command.CommandText = command.CommandText.ApplyCorrectYeKe();
```

در اصل نیازی به آن نیست؛ چون مقادیر ارسالی توسط پارامترها جابجا می‌شوند و در CommandText به صورت مستقیم حضور ندارند.

نویسنده: محمد زعفرانی
تاریخ: ۱۳۹۳/۰۱/۱۸ ۱۶:۴۸

سلام. چطور میتونم نسخه‌ی EF پروژه ام رو به EF6 ارتقاء بدم؟
اگر این ارتقاء رو انجام بدم به مشکلی در پروژه ام برنمیخورم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۸ ۱۶:۵۵

دو مطلب در این مورد پیشتر در سایت منتشر شده:
- ارتقاء به Entity framework 6 و استفاده از بانک‌های اطلاعاتی غیر از SQL Server
- بروز رسانی استفاده از SqlServer Compact در Entityframework 6.0
خلاصه هر دو مورد این است: یک فایل packages.config نیوگت را به پروژه‌هایی که ارجاعی به EF دارند اضافه کنید. بعد دستور update-package را صادر کنید.

نویسنده: همراز
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۱:۳۲

با تشکر از پست مفید جناب نصیری
من هم این کار را انجام داده ام در دو سطح برنامه و دیتابیس ...
ابتدا باید دیتا بیس را یکسان سازی کرد (یعنی ی و ک‌ها فقط از یک مدل باشند)

```
CREATE PROCEDURE [dbo].[spr_Admin_Replace_Ye_Ke_InAllTables]
AS
BEGIN
BEGIN TRAN
--ی--%u06CC
--ی--%u064A
--ک--%u06A9
--ک--%u0643
DECLARE @Ye_Farsi NCHAR(1),  @Ye_Arabi NCHAR(1), @Ke_Farsi NCHAR(1), @Ke_Arabi NCHAR(1)
SET @Ye_Farsi = NCHAR(0X06CC)
SET @Ye_Arabi = NCHAR(0X064A)
SET @Ke_Farsi = NCHAR(0X06A9)
SET @Ke_Arabi = NCHAR(0X0643)
--SELECT @Ye_Farsi, UNICODE(@Ye_Farsi) AS Ye_Farsi_Code, @Ye_Arabi, UNICODE(@Ye_Arabi) AS
Ye_Arabi_Code,@Ke_Farsi, UNICODE(@Ke_Farsi) AS Ke_Farsi_Code, @Ke_Arabi, UNICODE(@Ke_Arabi) AS
Ke_Arabi_Code

--SELECT * FROM sys.types
DECLARE xcur CURSOR FOR -- a cursor for string columns
SELECT sys.tables.name AS TableName, sys.columns.name AS ColumnName
FROM sys.tables INNER JOIN sys.columns ON sys.tables.object_id = sys.columns.object_id
WHERE sys.columns.system_type_id IN (35, 99, 167, 175, 231, 239)

OPEN xcur

DECLARE @SqlString nvarchar(1000), @TName nvarchar(255), @CName nvarchar(255), @ret int

FETCH NEXT FROM xcur INTO @TName, @CName

WHILE @@FETCH_STATUS = 0
BEGIN
BEGIN TRY
SET @SqlString = N'UPDATE ' + @TName + ' SET ' + @CName + ' = REPLACE( REPLACE(' + @CName + ',''' +
@Ye_Farsi + ''',''' + @Ye_Arabi + ''') ,''' +  @Ke_Farsi + ''',''' + @Ke_Arabi + ''')';
EXEC @ret = sp_executesql @SqlString
PRINT @ret
END TRY
BEGIN CATCH
PRINT @SqlString
PRINT ERROR_MESSAGE()
END CATCH

FETCH NEXT FROM xcur INTO @TName, @CName
END

CLOSE xcur
DEALLOCATE xcur

ROLLBACK TRAN
END
```

سپس داخل کد برنامه و هنگام ثبت، ویرایش و جستجو
ی و ک موجود در کلمات ورودی توسط کاربر را با ی و ک درست(همان‌ها که در دیتابیس هستند)، جایگزین کنیم، و بعد عمل مورد نظر را انجام دهیم.

```
public class YeKeLetters
    {
        public static char Ye_Farsi = '\x06CC'; // ی %u06CC
        public static char Ye_Arabi = '\x064A'; // ی %u064A
        public static char Ke_Farsi = '\x06A9'; // ک %u06A9
        public static char Ke_Arabi = '\x0643'; // ک %u0643
    }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۲:۳۶

توضیحات بیشتر در مورد اسکریپتی که ارسال کردند:
« مشکل ی و ک فارسی و عربی در یک دیتابیس اس کیوال سرور »

نویسنده: سالار خلیل زاده
تاریخ: ۱۳۹۳/۰۲/۱۵ ۸:۱۲

دلیل تغییراتی که در رشته رو میدید متوجه نشدم! استفاده از trim و بازگرداندن رشته خالی
به نظرم اینطوری بهتره:

```
public static string ApplyCorrectYeKe(this string data)
{
return string.IsNullOrWhiteSpace(data)
? data
: data.Replace(ArabicYeChar, PersianYeChar).Replace(ArabicKeChar, PersianKeChar);
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۵ ۱۰:۵

- بازگشت رشته خالی بجای نال: آشنایی با Defensive programming قسمت اول و دوم .
- حذف فواصل خالی: فواصل خالی ابتدا و انتهای رشته در خیلی از موارد نباید حضور داشته باشند. مثلا در ثبت نام فرق است بین «سالار» و « سالار ».

نویسنده: علی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۶:۲۰

سلام.
آیا اینجا می‌توان تمیزسازی HTML و مقابله با XSS را انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۸:۳۶

می‌توان.