عنوان: **نکاتی در مورد ELMAH**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱:۲۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, ELMAH

سفارشی سازی ایمیل ارسالی :

در مورد ELMAH(Erro Logging Module And Handlers) آقای نصیری چندین مطلب نوشته اند (+ و + و ...)
قبل از ارسال ایمیل توسط ELMAH رخدادی به نام Mailing اجرا (Raise) می‌شود. اگر برای این رخداد یک Event Handler ایجاد کنیم، می‌توانیم جزئیات مربوط به خطایی که قرار است ارسال شود را تغییر دهیم. به عنوان مثال می‌توانیم بگوئیم اگر Error ما از نوع Application Exception بود آنرا به آدرس دیگری ارسال کن و .... برای ایجاد یک Event Handler برای رخداد Mailing ابتدا فایل Global.asax رو به پروژه اضافه کنید و کد زیر را به آن اضافه کنید :

```
void ErrorMailModuleName_Mailing(object sender, Elmah.ErrorMailEventArgs e)
{

}
```

دقت داشته باشید که در کد فوق به جای ErrorMailModuleName نام HTTPModule را بنویسید، این ماژول در فایل Web.Config قرار دارد :

```
<httpModules>
      <add name="ErrorMail" type="Elmah.ErrorMailModule, Elmah" />
</httpModules>
```

که در کد فوق ErrorMail می‌باشد در نهایت :

```
void ErrorMail_Mailing(object sender, Elmah.ErrorMailEventArgs e)
    {

    }
```

به عنوان مثال در کد زیر اگر Error از نوع Application Exception بود Error Log به آدرس sir1afifi@gmail.com نیز ارسال می‌شود :

```
void ErrorMail_Mailing(object sender, Elmah.ErrorMailEventArgs e)
    {
        if (e.Error.Exception is ApplicationException)
        {
            e.Mail.To.Add("sir1afifi@gmail.com");
        }
    }
```

ذخیره Errorها در دیتابیس SQL SERVER :

ELMAH خطاهای Log شده را در جدولی با نام ELMAH_Error ذخیره می‌کند، اگر به داخل پوشه ELMAH توجه کرده باشید یک فایل با نام SQLServer.sql وجود دارد که حاوی اسکریپت مربوط به ساخت جدول فوق می‌باشد. با اجرای این اسکریپت جدول مربوطه همرا با سه SP با نام‌های ELMAH_LogError ، ELMAH_GetErrorXml ، ELMAH_GetErrorsXml ساخته می‌شوند. بعد از ساخت جدول مربوطه باید تگ زیر را در فایل Web.Config بنویسیم :

```
<errorLog type="Elmah.SqlErrorLog, Elmah"
            connectionStringName="..." />
```

connectionStringName هم نام کانکشن استرینگ را در این قسمت قرار می‌دهیم به عنوان مثال با داشتن کانکشن استرینگ زیر :

```
<connectionStrings>
    <add name="conn" connectionString="Data Source=.;Initial Catalog=test;User
ID=user1;Password=123456;"/>
  </connectionStrings>
```

به این صورت connectionStringName برابر با مقدار name یعنی conn می‌شود.

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۷/۰۳ ۱۱:۳۰

برای استفاده از Elmah در یک پروژه MVC باید پکیج Elmah.MVC رو دریافت کنیم یا همون Elmah?

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۳ ۱۱:۳۷

توضیحات بیشتر در اینجا

نویسنده: امیر
تاریخ: ۱۳۹۲/۰۵/۳۰ ۹:۳۱

با سلام؛ چند سوال:
چطوری بدون استفاده از ممبر شیپ نتونه هر کسی این آدرس زد این صفحه بیاد. یه جورایی یا پسورد ازش بگیره یا سطح دسترسی براش تعریف کنیم.

-2 در کدوم قسمت فایل وب کانفیگ تغییر بدم که اسم فایل elamh.axd تغییر کند مثلا بشه test.axd

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۳۰ ۹:۳۷

به این دو سؤال در مطالب پیشنیاز بحث جاری پاسخ داده شده‌اند:

معرفی ELMAH
مدیریت خطاها در یک برنامه ASP.NET MVC

نویسنده: امیر
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۰:۰

ممنون آقای نصیری .من خوندم ولی نگفته بودین کدوم قسمت رو تغییر بدیم تا نام elamh بشه. فقط گفته شد تغییر داده شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۰:۳۵

در همان مطلب معرفی عنوان شده، داریم:

```
<add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />
```

در اینجا مقدار path را بجای elmah.axd تغییر بدید به هر نام دیگری که علاقمند بودید.

نویسنده: امیر
تاریخ: ۱۳۹۲/۰۵/۳۱ ۱۲:۱۵

چطوری بدون استفاده از ممبر شیپ نتونه هر کسی این آدرس زد (www.yoursite.com/elmah.axd)این صفحه بیاد. یه جورایی یا پسورد ازش بگیره یا سطح دسترسی براش تعریف کنیم.چون سطح دسترسی من توسط کد نویسی هست

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۳۱ ۱۲:۲۲

در همان مطلب پیشنیاز در مورد location بحث شده، یا حداقل اشاره شده (و اگر کدهای آن ذکر نشده، چون یک مبحث مقدماتی است و نه تکمیلی). در یک مبحث *تکمیلی* فرض بر این است که شما حداقل یک فصل از مباحث ابتدایی Forms Authentication رو مطالعه کردید. مثلا می‌دونید تگ location را باید به چه صورتی تعریف کرد، یا چطور باید Role به آن اضافه کرد:

```
<location path=".............">
  <system.web>
    <authorization>
      <allow roles="Administrators" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

عنوان: **ELMAH و حملات XSS**
نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۳:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, ASP.Net MVC, ELMAH, xss

ASP.NET جهت مقابله با حملات XSS بطور پیش‌فرض از ورود تگ‌های HTML جلوگیری می‌کند و در صورتی که ورودی کاربر شامل این تگ‌ها باشد، HttpRequestValidationException صادر می‌گردد. لاگ کردن و بررسی این خطاها جهت آگاهی از وجود حمله بی اهمیت نیست. اما متأسفانه ELMAH که به عنوان معمول‌ترین ابزار ثبت خطاها کاربرد دارد این نوع Exceptionها را ثبت نمی‌کند. دلیل آن هم این است که ELMAH در رویه‌های درونی خود اقدام به خواندن ورودی‌های کاربر می‌کند و در این هنگام اگر ورودی کاربر نامعتبر باشد، Exception مذکور صادر می‌شود و فرصتی برای ادامه روند و ثبت خطا باقی نمی‌ماند. به هر حال ضروری است که این نقیصه را خودمان جبران کنیم. راه حل افزودن یک فیلتر سفارشی برای ثبت خطاها به شکل زیر است (ASP.NET MVC):

```
public class ElmahRequestValidationErrorFilter : IExceptionFilter
{
    public void OnException(ExceptionContext context)
    {
        if (context.Exception is HttpRequestValidationException)
           ErrorLog.GetDefault(HttpContext.Current).Log(new Error(context.Exception));
    }
}
```

فیلتر فوق باید در Global.asax معرفی شود:

```
public static void RegisterGlobalFilters (GlobalFilterCollection filters)
{
    filters.Add(new ElmahRequestValidationErrorFilter());
    filters.Add(new HandleErrorAttribute());
}
```

به این ترتیب HttpRequestValidationException هم بعد از این در سیستم ELMAH ثبت خواهد شد.

نظرات خوانندگان

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۳

با سلام، در چک لیست ASP.NET MVC مورد زیر وجود دارد
آیا مورد زیر کماکا معتبر است ؟
- فیلتر پیش فرض مدیریت خطاها حذف و بجای آن از ELMAH استفاده شود.

با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۶

- به سطر HandleErrorAttribute پیش فرض نیازی نیست (البته اگر تنظیمات وب کانفیگ درستی داشته باشید). در قسمت 16 سری MVC توضیح دادم. وجود آن سبب می‌شود که ELMAH اصلا کار نکند و خطای مدیریت نشده‌ای به آن ارجاع داده نشود (چون قبلا مدیریت شده).
- بله. همچنان ELMAH معتبر است. نکته فوق را هم اضافه کنید، کاملتر خواهد شد.

نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۸

سلام
این چک لیست توسط آقای نصیری تهیه شده بود. این مورد هم از نظر من معتبر است.
با تشکر

عنوان: **ثبت جزئیات استثناهای Entity framework توسط ELMAH**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۸ ۱۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Entity framework, MVC, ELMAH

در حین بروز استثناهای Entity framework، می‌توان توسط ابزارهای Logging متنوعی مانند ELMAH ، جزئیات متداول آن‌ها را برای بررسی‌های آتی ذخیره کرد. اما این جزئیات فاقد SQL نهایی تولیدی و همچنین پارامترهای ورودی توسط کاربر یا تنظیم شده توسط برنامه هستند. برای اینکه بتوان این جزئیات را نیز ثبت کرد، می‌توان یک IDbCommandInterceptor جدید را طراحی کرد.

کلاس EfExceptionsInterceptor

در اینجا نمونه‌ای از یک پیاده سازی اینترفیس IDbCommandInterceptor را مشاهده می‌کنید. همچنین طراحی یک متد عمومی که می‌تواند به جزئیات SQL نهایی و پارامترهای آن دسترسی داشته باشد، در اینترفیس IEfExceptionsLogger ذکر شده‌است.

```
public interface IEfExceptionsLogger
{
    void LogException<TResult>(DbCommand command,
        DbCommandInterceptionContext<TResult> interceptionContext);
}

using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;

namespace ElmahEFLogger
{
    public class EfExceptionsInterceptor : IDbCommandInterceptor
    {
        private readonly IEfExceptionsLogger _efExceptionsLogger;

        public EfExceptionsInterceptor(IEfExceptionsLogger efExceptionsLogger)
        {
            _efExceptionsLogger = efExceptionsLogger;
        }

        public void NonQueryExecuted(DbCommand command, DbCommandInterceptionContext<int>
interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

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

        public void ReaderExecuted(DbCommand command, DbCommandInterceptionContext<DbDataReader>
interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ReaderExecuting(DbCommand command, DbCommandInterceptionContext<DbDataReader>
interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ScalarExecuted(DbCommand command, DbCommandInterceptionContext<object>
interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

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

تهیه یک پیاده سازی سفارشی از IEfExceptionsLogger توسط ELMAH

اکنون که ساختار کلی IDbCommandInterceptor سفارشی برنامه مشخص شد، می‌توان پیاده سازی خاصی از آن‌را جهت استفاده از ELMAH به نحو ذیل ارائه داد:

```
using System;
using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;
using Elmah;

namespace ElmahEFLogger.CustomElmahLogger
{
    public class ElmahEfExceptionsLogger : IEfExceptionsLogger
    {
        /// <summary>
        /// Manually log errors using ELMAH
        /// </summary>
        public void LogException<TResult>(DbCommand command,
            DbCommandInterceptionContext<TResult> interceptionContext)
        {
            var ex = interceptionContext.OriginalException;
            if (ex == null)
                return;

            var sqlData = CommandDumper.LogSqlAndParameters(command, interceptionContext);
            var contextualMessage = string.Format("{0}{1}OriginalException:{1}{2} {1}", sqlData,
Environment.NewLine, ex);


            if (!string.IsNullOrWhiteSpace(contextualMessage))
            {
                ex = new Exception(contextualMessage, new ElmahEfInterceptorException(ex.Message));
            }

            try
            {
                ErrorSignal.FromCurrentContext().Raise(ex);
            }
            catch
            {
                ErrorLog.GetDefault(null).Log(new Error(ex));
            }
        }
    }
}
```

در اینجا شیء Command به همراه SQL نهایی تولید و پارامترهای مرتبط است. همچنین
interceptionContext.OriginalException جزئیات عمومی استثنای رخ داده را به همراه دارد. می‌توان این اطلاعات را پس از اندکی نظم بخشیدن، به متد Raise مربوط به ELMAH ارسال کرد تا جزئیات بیشتری از استثنای رخ داده شده، در لاگ‌های آن ظاهر شوند.

استفاده از ElmahEfExceptionsLogger جهت طراحی یک Interceptor عمومی

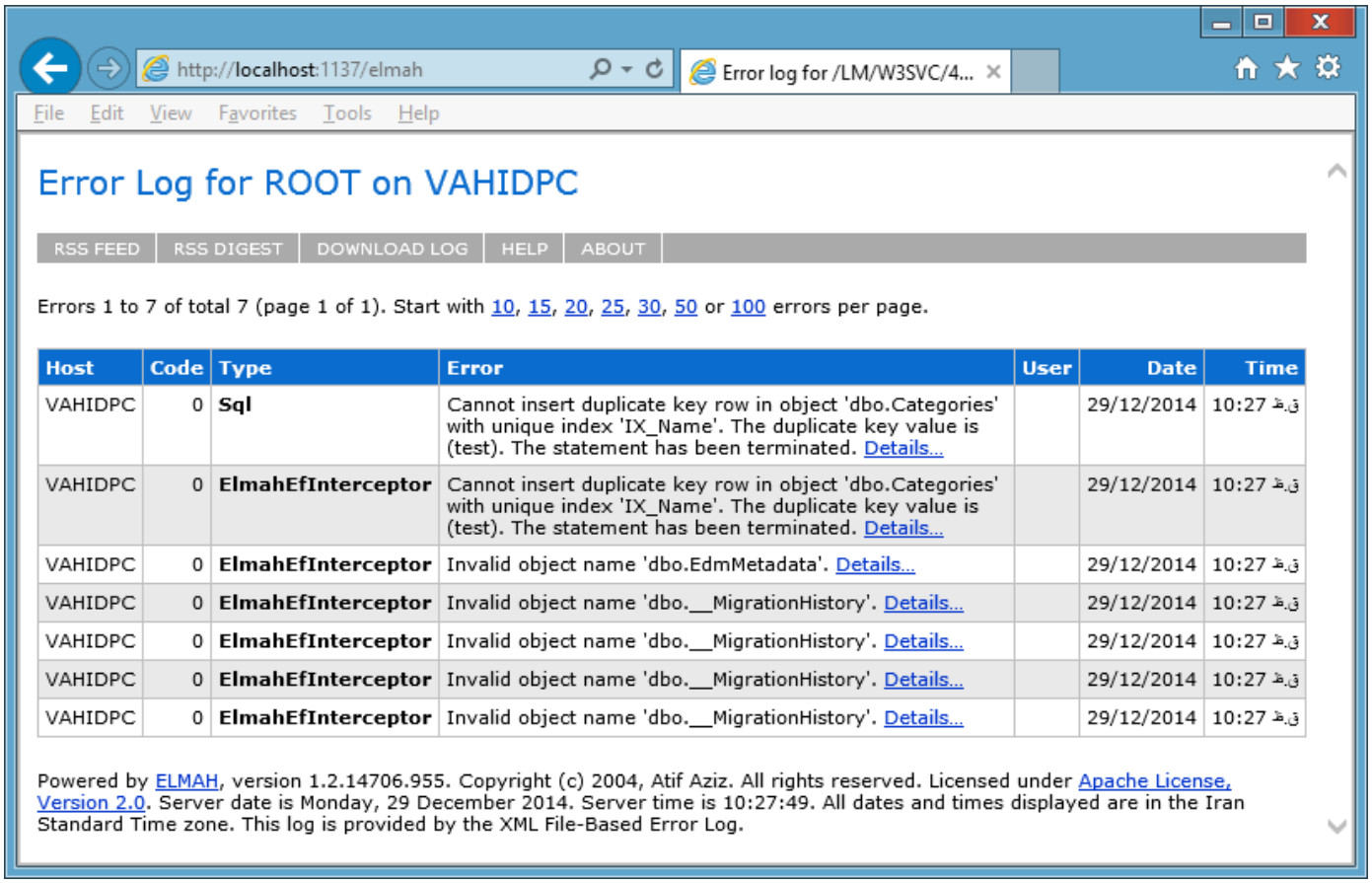
```
public class ElmahEfInterceptor : EfExceptionsInterceptor
    {
        public ElmahEfInterceptor()
            : base(new ElmahEfExceptionsLogger())
        { }
    }
```

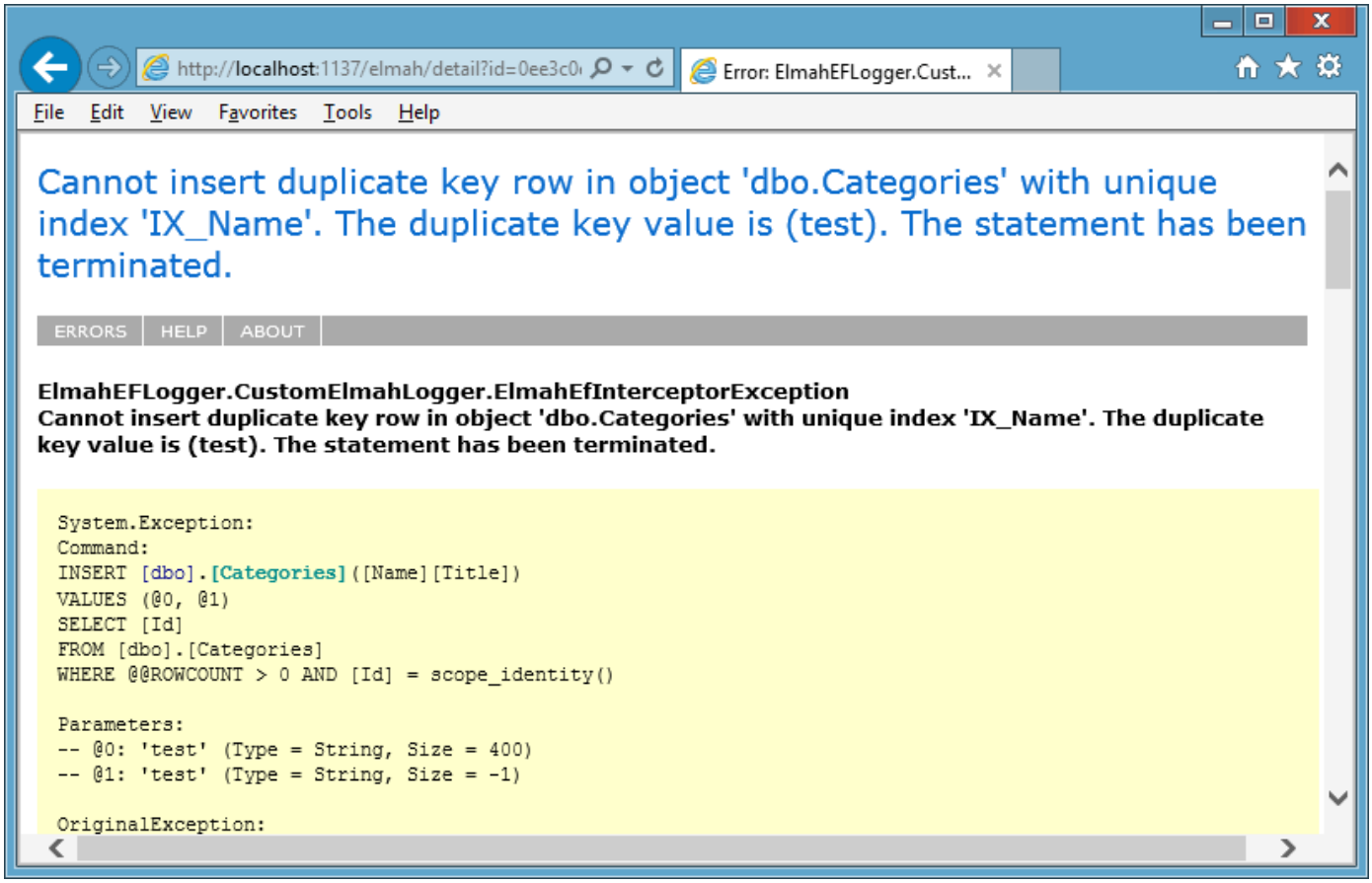
برای استفاده از ElmahEfExceptionsLogger و تهیه یک Interceptor عمومی، می‌توان با ارث بری از کلاس Interceptor ابتدای بحث شروع کرد و وهله‌ای از ElmahEfExceptionsLogger را به سازنده‌ی آن تزریق نمود (یکی از چندین روش ممکن). سپس برای

استفاده از آن کافی است به ابتدای متد Application_Start فایل Global.asax.cs مراجعه و در ادامه سطر ذیل را اضافه نمود:

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

پس از آن جزئیات کلیه استثناهای EF در لاگ‌های نهایی ELMAH به نحو ذیل ظاهر خواهند شد:





کدهای کامل این پروژه را از اینجا می‌توانید دریافت کنید:

ElmahEFLogger

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۰:۴۷

با سلام و تشکر از مطلب شما

در صورت استفاده از پکیج بالا باید فیلتر کلی که در پست آموزش MVC قسمت 16 توضیح دادین برای ثبت استثناهای elmah حذف بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۱:۲۵

خیر. کار این Interceptor صرفا لاگ کردن سراسری و نامرئی ریز جزئیات استثناهای EF در کل برنامه است. استثناهای رخ داده، در ادامه از فیلتر ELMAH اصلی تنظیم شده رد خواهند شد و یکبار هم در آنجا لاگ می‌شوند. این مورد چون منحصر به EF نیست، بنابراین ضرورتی به حذف آن هم نیست.