

## مدیریت خطاها در یک برنامه ASP.NET MVC

## استفاده از فیلتر HandleError

یکی از فیلترهای توکار ASP.NET MVC به نام HandleError، می‌تواند کار هدایت کاربر را به یک صفحه‌ی خطای عمومی، در حین بروز استثنایی در برنامه، انجام دهد. برای آزمایش آن یک برنامه خالی جدید ASP.NET MVC را آغاز کنید. سپس یک کنترلر جدید را با محتوای زیر به آن اضافه نمایید:

```
using System;
using System.Web.Mvc;

namespace MvcApplication13.Controllers
{
    public class HomeController : Controller
    {
        [HandleError]
        public ActionResult Index()
        {
            throw new InvalidOperationException();
            return View();
        }
    }
}
```

در اینجا جهت آزمایش برنامه، به عمد یک استثنای دستی را صادر می‌کنیم. برای آزمایش برنامه هم نیاز است آن را خارج از دیباگر VS.NET اجرا کرد (آدرس برنامه را مستقیماً خارج از VS.NET در یک مرورگر وارد کنید). همچنین یک سطر زیر را نیز لازم است به فایل web.config برنامه اضافه نمایید:

```
<system.web>
  <customErrors mode="On" />
```

اکنون اگر برنامه را خارج از مرورگر اجرا کنید، با توجه به استفاده از ویژگی HandleError و همچنین بروز یک استثنا در متد Index، خودبخود صفحه Views\Shared\Error.cshtml به کاربر نمایش داده خواهد شد. در غیراینصورت صفحه زرد رنگ پیش فرض خطای ASP.NET به کاربر نمایش داده می‌شود که محتوای آن‌ها بیشتر برای برنامه نویسی‌ها مناسب است و نه کاربران نهایی سیستم.

اگر علاقمند باشید که این ویژگی به صورت خودکار به تمام متدهای کنترلرهای برنامه اعمال شود، کافی است یک سطر زیر را به متد Application\_Start فایل Global.asax.cs اضافه نمایید:

```
GlobalFilters.Filters.Add(new HandleErrorAttribute());
```

البته نیازی به انجام اینکار نیست زیرا اگر به متد `RegisterGlobalFilters` فایل `Global.asax.cs` دقت کنیم، اینکار پیشتر توسط قالب پیش فرض `VS.NET` انجام شده است. فقط برای فعال سازی آن نیاز است تگ `customErrors` در فایل وب کانفیگ برنامه مقدار دهی و تنظیم شود.

### استفاده از صفحه خطای سفارشی دیگری بجای فایل `Error.cshtml`

امکان تنظیم نمایش صفحه خطای سفارشی دیگری نیز وجود دارد. برای مثال استفاده از فایل `Views\Shared\CustomErrorView.cshtml`:

```
[HandleError(View = "CustomErrorView")]
```

### استفاده از صفحات خطای متفاوت به ازای استثنای مختلف

می توان فیلتر `HandleError` را تنها به یک نوع استثنای خاص محدود کرد. همچنین امکان استفاده از چندین ویژگی `HandleError` برای یک متد نیز وجود دارد:

```
[HandleError(ExceptionType = typeof(NullReferenceException), View = "ErrorHandling")]
```

### دسترسی به اطلاعات استثناء در صفحه نمایش خطاها

زمانیکه برنامه به صفحه خطا هدایت می شود، نوع `Model` آن `System.Web.Mvc.HandleErrorInfo` می باشد:

```
@model System.Web.Mvc.HandleErrorInfo
@{
    ViewBag.Title = "DbError";
}
<h2>An Error Has Occurred</h2>
@if (Model != null)
{
    <p>@Model.Exception.GetType().Name<br />
    thrown in @Model.ControllerName @Model.ActionName</p>
}
```

البته این نکته را صرفا به عنوان اطلاعات عمومی در نظر داشته باشید. زیرا اگر قرار باشد مجددا اصل استثناء را نمایش دهیم، همان صفحه زرد رنگ `ASP.NET` شاید بهتر باشد.

### استفاده از تگ `customErrors` در فایل `Web.config` برنامه

ویژگی حالت تگ `customErrors` در فایل `web.config` برنامه، سه مقدار را می تواند بپذیرد:  
 الف) `Off`: صفحه زرد رنگ معرفی خطای `ASP.NET` را به همراه تمام اطلاعات مرتبط با استثنای رخ داده نمایش می دهد.  
 ب) `RemoteOnly`: همان حالت الف است با این تفاوت که صفحه خطا را فقط در کامپیوتری که وب سرور بر روی آن نصب است

نمایش خواهد داد.

ج) On : یک صفحه خطای سفارشی شده را نمایش می‌دهد.

بنابراین هیچگاه از حالت Off استفاده نکنید. زیرا خطاهای نمایش داده شده، علاوه بر برنامه نویس، برای مهاجم به یک سایت نیز بسیار دلپذیر است!

حالت RemoteOnly در زمان توسعه برنامه توصیه می‌شود.

حالت On حین توزیع برنامه باید بکار گرفته شود.

### مدیریت خطاهای رخ داده خارج از MVC Pipeline

HandleErrorAttribute تنها استثنای رخ داده داخل ASP.NET MVC Pipeline را مدیریت می‌کند (یا خطاهایی از نوع 500). اگر این نوع استثناها خارج از آن رخ دهند مثلاً فایلی یافت نشود (خطای 404) و امثال آن، باید به روش زیر عمل کرد:

```
<customErrors mode="On" defaultRedirect="error">
  <error statusCode="404" redirect="error/notfound" />
  <error statusCode="403" redirect="error/forbidden" />
</customErrors>
```

در اینجا اگر فایلی یافت نشد، کاربر به کنترلری به نام error و متدی به نام notfound هدایت خواهد شد. بنابراین نیاز به کنترلر زیر وجود دارد؛ به علاوه به ازای هر متد هم یک View متناظر باید اضافه شود (کلیک راست روی نام متد و انتخاب گزینه افزودن View جدید).

```
using System.Web.Mvc;

namespace MvcApplication13.Controllers
{
    public class ErrorController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult NotFound()
        {
            return View();
        }

        public ActionResult Forbidden()
        {
            return View();
        }
    }
}
```

برای آزمایش این قسمت، برنامه را اجرا کرده و سپس مثلاً آدرس غیرموجود <http://localhost/xyz> را وارد کنید.

### استفاده از فیلتر HandleError اجباری نیست

در همین قسمت قبل پس از افزودن customErrors و defaultRedirect آن که به نام یک کنترلر اشاره می‌کند، کلیه فیلترهای HandleError اضافه شده به برنامه را حذف کنید. سپس برنامه را خارج از محیط VS.NET اجرا کنید. باز هم متد Index کنترلر Error اجرا خواهد شد. به عبارتی الزامی به استفاده از فیلتر HandleError نیست و به کمک مقدار دهی صحیح تگ customErrors، کار نمایش خودکار صفحه سفارشی خطاها به کاربر انجام خواهد شد.

البته بدیهی است که گزینه‌های نمایش یک View خاص به ازای استثنایی ویژه، یکی از مزیت‌های استفاده از فیلتر `HandleError` می‌باشد که امکان تنظیم آن در فایل `web.config` وجود ندارد.

### ثبت اطلاعات استثنای رخ داده به کمک ELMAH

نمایش صفحه‌ی خطای سفارشی به کاربر، یکی از موارد ضروری تمام برنامه‌های ASP.NET است، اما کافی نیست. ثبت اطلاعات جزئیات استثنای رخ داده در طول زمان می‌تواند به بالا بردن کیفیت برنامه به شدت کمک کند. برای این منظور می‌توان همانند سابق از متد `Application_Error` قابل تعریف در فایل `Global.asax.cs` کمک گرفت؛ اما با وجود افزونه‌ای به نام [ELMAH](#) اینکار اتلاف وقت است و اصلاً توصیه نمی‌شود. همچنین به کمک ELMAH می‌توان مشکلات را تبدیل به ایمیل‌های خودکار کرد یا از آن‌ها فید RSS درست نمود.

برای دریافت ELMAH یا به سایت اصلی آن مراجعه نمائید و یا به کمک [NuGet](#) هم به سادگی قابل دریافت است. پس از دریافت، ارجاعی را به اسمبلی آن (`Elmah.dll`) اضافه نمائید. در ادامه فایل `web.config` برنامه را گشوده و چند سطر زیر را به آن در قسمت `configuration` اضافه کنید:

```
<configuration>
  <configSections>
    <sectionGroup name="elmah">
      <section name="security" requirePermission="false" type="Elmah.SecuritySectionHandler, Elmah"/>
      <section name="errorLog" requirePermission="false" type="Elmah.ErrorLogSectionHandler, Elmah"/>
      <section name="errorMail" requirePermission="false" type="Elmah.ErrorMailSectionHandler, Elmah"/>
      <section name="errorFilter" requirePermission="false" type="Elmah.ErrorFilterSectionHandler, Elmah"/>
      <section name="errorTweet" requirePermission="false" type="Elmah.ErrorTweetSectionHandler, Elmah"/>
    </sectionGroup>
  </configSections>
```

سپس ذیل قسمت `appSettings`، تنظیمات پروایدر ذخیره سازی اطلاعات آن‌را وارد نمائید. مثلاً در اینجا از فایل‌های XML برای ذخیره سازی اطلاعات استفاده خواهد شد (که امن‌ترین حالت ممکن است؛ از این لحاظ که اگر بانک اطلاعاتی را انتخاب کنید، ممکن است مشکل اصلی از همانجا ناشی شده باشد. بنابراین خطایی ثبت نخواهد شد. همچنین در این حالت نیازی به سایر DLL‌های همراه ELMAH هم نیست). در اینجا مسیر ذخیره سازی اطلاعات در پوشه `app_data/errorslog` تنظیم شده است:

```
<elmah>
  <security allowRemoteAccess="1"/>
  <errorLog type="Elmah.XmlFileErrorLog, Elmah" logPath="~/App_Data/ErrorsLog"/>
</elmah>
```

در ادامه در قسمت `system.web`، دو تعریف زیر را اضافه نمائید. به این ترتیب امکان دسترسی به آدرس `http://server/elmah.axd` مهیا می‌گردد:

```
<httpModules>
  <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
</httpModules>
<httpHandlers>
  <add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah"/>
</httpHandlers>
```

البته برای IIS7 تنظیمات ذیل نیز باید اضافه شوند:

```
<system.webServer>
  <validation validateIntegratedModeConfiguration="false"/>
  <modules runAllManagedModulesForAllRequests="true">
    <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
  </modules>
  <handlers>
    <add name="Elmah" verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah"/>
  </handlers>
</system.webServer>
```

و به این ترتیب تنظیمات اولیه ELMAH به پایان می‌رسد (و با ASP.NET Web forms هیچ تفاوتی ندارد). مرحله بعد، تنظیمات مسیریابی ASP.NET MVC است برای اینکه آدرس `http://server/elmah.axd` را وارد سیستم پردازشی خود نکند. البته اینکار پیشتر انجام شده است:

```
public static void RegisterRoutes(RouteCollection routes)
{
    //routes.IgnoreRoute("elmah.axd");
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
}
```

بنابراین همین تنظیمات، به همراه قالب پیش فرض یک پروژه جدید ASP.NET MVC برای استفاده از ELMAH کفایت می‌کند. اکنون پروژه جاری را یکبار دیگر خارج از VS.NET اجرا کرده و سپس به مسیر `http://localhost/elmah.axd` جهت مشاهده خطاهای لاگ شده به همراه جزئیات کامل آن‌ها مراجعه کنید.

### مشکل: استثنای برنامه توسط ELMAH لاگ نمی‌شوند!

فیلتر `HandleError` با ELMAH سازگار نیست. زیرا با استفاده از آن، متدهای کنترلرها به صورت خودکار داخل یک `try/catch` اجرا شده و به این ترتیب استثنای رخ داده، مدیریت گردیده و به ELMAH هدایت نمی‌شوند. بنابراین نیاز است به متد `RegisterGlobalFilters` فایل `Global.asax.cs` مراجعه کرده و سطر زیر را حذف کنید:

```
filters.Add(new HandleErrorAttribute());
```

و یا اگر قصد نداشتید اینکار را انجام دهید، می‌توان به نحو زیر نیز مشکل را حل کرد:

```
using System.Web.Mvc;
using Elmah;

namespace MvcApplication13.CustomFilters
{
    public class ElmahHandledErrorLoggerFilter : IExceptionHandler
    {
        public void OnException(ExceptionContext context)
        {
            if (context.ExceptionHandled)
                ErrorSignal.FromCurrentContext().Raise(context.Exception);
            // all other exceptions will be caught by ELMAH anyway
        }
    }
}
```

در اینجا یک فیلتر سفارشی به برنامه اضافه شده است تا خطاهای مدیریت شده برنامه (خطاهای مدیریت شده توسط فیلتر

HandleError توکار) را به موتور ELMAH هدایت کند. سایر خطاهای مدیریت نشده به صورت خودکار توسط ELMAH ثبت خواهند شد و نیازی به انجام کار اضافی در این مورد نیست. سپس این فیلتر جدید را به صورت سراسری تعریف کنید:

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

ترتیب این‌ها هم مهم است. ابتدا باید ElmahHandledErrorLoggerFilter معرفی شود.

### تذکر مهم!

حین استفاده از ELMAH یک نکته را فراموش نکنید: اگر allowRemoteAccess آن را به عدد 1 تنظیم کرده‌اید، به هیچ عنوان از نام پیش فرض elmah.axd استفاده نکنید (هر نام اختیاری دیگری را که علاقمند بودید و به سادگی قابل حدس زدن نبود، در فایل web.config وارد کنید).

### خلاصه بحث

- 1- در ASP.NET MVC نیازی نیست تا متدهای کنترلرها را با try/catch شلوغ کنید.
- 2- حتما قسمت customErrors فایل وب کانفیگ برنامه را دهی کنید (این مورد را به چک لیست اجباری تهیه یک برنامه ASP.NET MVC اضافه کنید).
- 3- استفاده از فیلتر HandleError اختیاری است. اگر از قابلیت فیلتر کردن استثناهای ویژه آن استفاده نمی‌کنید، مقدار دهی customErrors وب کانفیگ برنامه هم همان کار را انجام می‌دهد.
- 4- برای ثبت جزئیات دقیق استثناهای رخ داده در برنامه، از ELMAH استفاده کنید و بی‌جهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید.

### مطالب مشابه

[معرفی ELMAH](#)

[ثبت استثناهای مدیریت شده توسط ELMAH](#)

## نظرات خوانندگان

نویسنده: Ahmadi rad  
تاریخ: ۲۱:۴۵:۴۰ ۱۳۹۱/۰۱/۲۸

سلام علیکم  
از غیرتی که در یاد آموزی مطالب در سطح خود به زبان فارسی دارید از شما سپاسگزاریم.  
باشد که ما و دیگران نیز همتی کنیم و با اندوخته ناچیز خود اینترنت را برای فارسی زبانان جویای دانش پر بار تر کنیم که امر امر پسندیده ای است.

نویسنده: Salehi  
تاریخ: ۱۰:۲۱:۳۱ ۱۳۹۱/۰۱/۳۰

با سلام و تشکر واقعی از این مطالب مفید.  
اگر از روش اول برای ثبت استثناها توسط Elmah استفاده شود، یعنی حذف یک سطر از فایل Global.asax.cs، فقط استثناها ثبت می شود، یا صفحه خطایی هم نمایش داده می شود؟

نویسنده: وحید نصیری  
تاریخ: ۰۹:۲۳:۲۹ ۱۳۹۱/۰۲/۰۱

بستگی داره cutomErrors رو مقدار دهی کردید یا خیر. اگر بله، مطابق مقدار دهی آن رفتار خواهد شد؛ مثلا یک صفحه سفارشی نمایش داده می شود. ولی خطایی از دست ELMAH در نمیره!

نویسنده: فرید صالحی  
تاریخ: ۱:۰۹ ۱۳۹۱/۰۴/۰۲

1- وقتی در web.config مقدار debug برابر با true هستش اداره خطا انجام نمیشه. آیا وقتی سایت publish شد و روی هاست قرار گرفت این مساله برطرف میشه یا حتما باید "debug=false" رو به خاطر داشته باشیم؟

2- با وجود اینکه خطا صادر میشه اما رویداد application\_error فراخوانی نمیشه. راه حل هایی برای این مساله گفته شده اما جواب نداد. ضمن اینکه یک فیلتر سفارشی با توسعه iexceptionfilter هم ایجاد و تغییرات لازم در global.asax انجام شد اما onexception در اونجا هم فراخوانی نمی شد.

3- آیا راهی وجود داره که برای هر area بشه مسیر اختصاصی برای خطاهای 403 و 404 معرفی کرد؟ چون در بخش customerrors یک مسیر کلی داده میشه

4- این ادیتوری که شما اینجا استفاده کردید خیلی سبک و قویه. اگر میشه معرفی کنید!

نویسنده: شاهین کیاست  
تاریخ: ۴:۳۹ ۱۳۹۱/۰۴/۰۲

ادیتور <http://redactorjs.com> هست.

نویسنده: وحید نصیری  
تاریخ: ۸:۴۹ ۱۳۹۱/۰۴/۰۲

- با debug=false هم انجام میشه. برنامه رو خارج از ویژوال استودیو اجرا کنید. بحث دیباگ کد در VS.NET با مدیریت خطا

توسط ASP.NET متفاوت است. debug=false برنامه ASP.NET رو تبدیل به حالت Release می‌کنه اما به این معنا نیست که مکانیزم‌های ASP.NET رو کلاً از کار می‌اندازه. مصرف حافظه برنامه کمتر میشه. debug symbols از اسمبلی نهایی حذف خواهند شد. کامپایلر روی کد نهایی بهینه‌سازی‌هایی در جهت بالابردن سرعت انجام می‌ده.

- در MVC اگر از فیلتر Handle Error استفاده بشه application\_error فراخوانی نخواهد شد. به علاوه با وجود ELMAH ضرورتی به استفاده از Handle error و application\_error نیست. توضیح دادم در متن فوق.

ضمن اینکه دیباگ کار شخصی افراد نیاز به بودن پروژه آن‌ها و بررسی آن‌ها در یک انجمن مرتبط یا توسط یک مشاور دارد. این جملات مبهم و کلی «من یه کاری یه جایی کردم، یکی یه چیزی گفت، برای من کار نکرد» ارزش یا امکان بررسی ندارند.

- custom errors قرار گرفته در ریشه سایت بر روی کل سایت اعمال می‌شود (و تمام برنامه‌هایی که در زیر پوشه‌های آن هستند). در فایل‌های کانفیگ ASP.NET مباحث ارث بری وجود دارند. برای لغو آن یکی از راه‌ها استفاده از تگ location است، مثلاً:

```
<location path="MyAreaName">
  <system.web>
    <customErrors mode="On" defaultRedirect="/MyAreaName/error" />
  </system.web>
</location>
```

نویسنده: شهروز جعفری  
تاریخ: ۱۷:۵۰ ۱۳۹۱/۰۴/۰۲

مثل اینکه پولی هستش

نویسنده: وحید نصیری  
تاریخ: ۱۷:۵۶ ۱۳۹۱/۰۴/۰۲

نه. برای استفاده در سایت‌های غیرتجاری، رایگان است. من از همان نسخه‌ای که در سایت قرار داده برای دانلود، استفاده می‌کنم. نسخه کاملی است و قابلیت آپلود هم دارد. البته مثال‌های آن با PHP است که تبدیل آن به دات نت کار ساده‌ای است. به علاوه نسخه رایگان آن پشتیبانی ندارد که مهم نیست.

ضمن اینکه درخواست من این است: لطفاً بحث جاری مقابله با خطاها را تغییر جهت دهید. با تشکر.

نویسنده: hasani  
تاریخ: ۱۷:۴۸ ۱۳۹۱/۰۵/۲۳

با سلام و تشکر از مطالب شما

1) ELMAH با [Elmah.MVC](#) فرقی دارد؟ آیا برای پروژه‌های ASP.Net MVC بهتر است از [Elmah.MVC](#) استفاده شود یا فرقی نمی‌کند.

2) در حالتی که پروژه عمومی نیست مثل سیستم حسابداری، آیا بهتر است مسیر <http://localhost/elmah.axd> پس از لاگین قابل مشاهده باشد یا پیشنهاد نمی‌کنید.

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

ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۸:۵ ۱۳۹۱/۰۵/۲۳

- فرقی نمی‌کنه. Elmah.MVC توضیحات این قسمت رو خلاصه کرده.



- اگر به استفاده کنندگان اطمینان دارید، مهم نیست.

نویسنده: امیر  
تاریخ: ۱۵:۲۰ ۱۳۹۱/۰۶/۰۲

سلام

اگر تو سایتمون از area استفاده کرده باشیم تو فیلتر HandleError موقع انتخاب view مورد نظرمون چه جوری باید نام view مورد نظرمون را بنویسیم

نویسنده: وحید نصیری  
تاریخ: ۱۶:۴۳ ۱۳۹۱/۰۶/۰۲

همیشه امکان مسیر دهی کامل وجود دارد:

```
[HandleError(View = "~/Views/SomeLocation/Index.cshtml")]
```

نویسنده: ناصر طاهری  
تاریخ: ۱۶:۵۶ ۱۳۹۱/۱۲/۱۸

سلام

شما در متن گفتید که "البته برای IIS7 تنظیمات ذیل نیز باید...".  
در IIS8 هم این تنظیمات باید باشه؟ چون به این خط :

```
<modules runAllManagedModulesForAllRequests="true">
```

خطا داد.

نویسنده: وحید نصیری  
تاریخ: ۱۷:۳۹ ۱۳۹۱/۱۲/۱۸

- این «خطا داد» یعنی چی؟

- از [جستجوی گوگل](#) استفاده کنید برای IIS8 و همین تنظیم فوق. اولین لینک [پاسخ رو داره](#).

نویسنده: ناصر طاهری  
تاریخ: ۱۷:۵۹ ۱۳۹۱/۱۲/۱۸

پس اجازه دسترسی برای انجام این تنظیمات رو ندارم.

هاست رایگان استفاده کردم با این آدرس : <https://somee.com/FreeAspNetHosting.aspx>  
ممنون.

نویسنده: leila  
تاریخ: ۹:۳ ۱۳۹۲/۱۱/۲۲

با تشکر فراوان؛ شما در مقاله اشاره کردید

1- در ASP.NET MVC نیازی نیست تا متدهای کنترلرها را با try/catch شلوغ کنید.

با توجه به [این مقاله](#) HandleErrorAttribute چهار محدودیت دارد :

1. Not support to log the exceptions

2. Doesn't catch HTTP exceptions other than 500
3. Doesn't catch exceptions that are raised outside controllers
4. Returns error view even for exceptions raised in AJAX call

درباره مورد 2 شما لطف کردید توضیح دادید درباره بقیه موارد چطور؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۱/۲۲ ۹:۲۳

نکته مهم و خلاصه مطلب جاری این است: از `HandleErrorAttribute` استفاده نکنید و از `ELMAH` استفاده کنید. تمام توضیحات آن جهت رسیدن به چهارمین موردی بوده که در پایان مطلب ذکر شده. جهت تکرار مجدد آن «... برای ثبت جزئیات دقیق استثنای رخ داده در برنامه، از [ELMAH](#) استفاده کنید و بی‌جهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید...»