کنترل خطا در ASP.NET

مترجم: سید احمد حسینی

asm_3010@yahoo.com

webography.ir

فهرست مطالب

پیشگفتار ۰

مقدمه ۱

کلاس Exception ۱

سلسه مراتب ارث بری استثنا ۱

سلسله مراتب کنترل استثنا ۲

کنترل خطا در سطح نرم افزار ۲

کنترل خطا در سطح صفحه ۳

کنترل خطا در سطح کد ۳

افزودن قابلیت ثبت خطا ۴

افزودن صفحه خطا ۵

نمایش پیام استثنای کنترل نشده در نرم افزار ۶

بروز رسانی پیکربندی ۶

اجرای برنامه ۷

افزودن یک خطای عمدی ۸

اجرای برنامه ۸

افزودن کنترل خطا در سطح نرم افزار ۸

اجرای برنامه ۹

افزودن کنترل خطا در سطح صفحه ۱۰

اجرای برنامه ۱۰

افزودن کنترل خطا در سطح کد ۱۱

بررسی اطلاعات خطاهای ثبت شده ۱۲

پیام های خطا ۱۲

به نام خدا

پیشگفتار

رخداد و اتفاق خطا جزئی از کار برنامه نویسی است.خطاها به روش های متفاوتی ایجاد می شوند.اغلب خطاها در هنگام ایجاد و ویرایش کدها ی برنامه اتفاق می افتند و معمولا توسط برنامه نویس کشف و برطرف می شوند.اما برخی دیگر نیز در هنگام تعامل کاربر نهایی با برنامه ایجاد می شوند.یک برنامه نویس خوب باید همواره سعی نماید تا از وقوع خطاهای احتمالی جلوگیری نماید و همچنین باید مکانیزم هایی را به کار گیرد تا در صورت ایجاد خطا بتواند آنها را ثبت و پیگیری و درنهایت برطرف نماید.نکته مهم تر اینکه خطای ایجاد شده باید به گونه ایی کنترل گردد که کمترین تاثیر را بر روند کاری برنامه و کاربر داشته باشد.

با توجه به اهمیت کنترل خطا در برنامه و به خصوص در برنامه های تحت وب و با توجه به مطالب بالا برای این متن ترجمه فصل "ASP.NET Error Handling" از کتاب Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013 نوشته Erik Reitan به عنوان منبع برای این موضوع انتخاب،ترجمه و تقدیم شما عزیزان گردیده است.

سید احمد حسینی

asm_3010@yahoo.com

webography.ir

در این متن تلاش شده است تا مفاهیم کنترل خطا و ثبت خطا آموزش داده شود.کنترل خطا به نرم افزار اجازه می دهد تا به خوبی خطاها را کنترل نماید و پیام های متناسب با هر خطا را نمایش دهد.همچنین ثبت خطاها به شما اجازه می دهد تا خطاهای ایجاد شده را یافته و برطرف نمایید.

در این متن از یک پروژه ASP.NET Forms استفاده شده است.

چیزهایی که شما فرا خواهید گرفت:

- نحوه افزودن مکانیزم های سراسری کنترل خطا به پیکر بندی نرم افزار
- نحوه افزودن مکانیزم های کنترل خطا در سطوح نرم افزاری،صفحات و کد
- نحوه ثبت خطا
- نحوه نمایش پیام ها

مقدمه

ASP.NET باید توانایی کنترل خطاهایی که در حین اجرا رخ می دهند را با استفاده از یک روش پایدار داشته باشد.ASP.NET از Common Language Runtime (CLR) استفاده می کند،که یک روش یکسان را برای آگاه کردن نرم افزارها از خطاها مهیا می کند.زمانی که یک خطا رخ می دهد یک استثنا ایجاد می گردد.یک استثنا خطا،شرایط یا هر رفتار غیرمنتظره ایی است که نرم افزار با آن مواجه می گردد.

در چارچوب .NET. یک استثنا یک شی است که از کلاس System. Exception ارث می برد.در هر بخشی از کد که مشکلی وجود داشته باشد یک استثنا ایجاد می گردد.خطاها به قسمتی ارجاع داده می شوند که برنامه کدی را برای کنترل استثنا مهیا کرده است.اگر برنامه خطا را کنترل نکند،مرورگر ناچار به نمایش جزییات خطاست.

به عنوان یک راهکار مناسب،خطاها را با استفاده از بلوک های try/catch/finally کنترل نمایید.سعی کنید از این روش در مکان هایی استفاده نمایید که کاربر را قادر سازد تا مشکل را برطرف نماید.اگر مکان استفاده از این روش از محل رخ دادن خطا دور باشد،فراهم نمودن اطلاعات مورد نیاز برای برطرف نمودن خطا مشکل می گردد.

کلاس Exception

کلاس Exception یک کلاس پایه است که استثناها از آن ارث می برند.اکثر استثناها نمونه ایی مشتق شده از این کلاس هستند.به عنوان مثال کلاس SystemException.

کلاس Exception خصوصیاتی مانند Message، Inner Exception،StackTrace را دارا می باشد که اطلاعات مشخصی را در مورد خطای رخ داده مهیا می کند.

سلسه مراتب ارث بری استثنا

CLR یک مجموعه از استثناهای مشتق شده از کلاس SystemException را دارا می باشد که در زمان مواجهه با استثنا ایجاد می گردند.اکثر کلاس هایی که از کلاس SystemException ارث برده اند مانند IndexOutOfRangeException، ArgumentNullException اعضای دیگری ندارند.بنابراین اکثر اطلاعات مهم در سلسله مراتب استثنا،نام استثنا و اطلاعاتی که در داخل استثنا وجود دارد یافت می شود.

سلسله مراتب کنترل استثنا

در ASP.NET Forms Application می توان استثناها را بر اساس یک سلسله مراتب خاص کنترل نمود. استثناها را در سطوح زیر کنترل می شوند:

- سطح نرم افزاری
- سطح صفحه
- سطح کد

زمانی که یک خطا را در سطح نرم افزار کنترل می نماییم ،اطلاعات تکمیلی در مورد استثنا که از کلاس Exception ارث بری شده است می تواند بازیابی و به کاربر نمایش داده شود.علاوه بر سطوح ذکر شده شما می توانید استثناها را از طریق ماژول HTTP و یا با استفاده از IIS کنترل نمایید.

کنترل خطا در سطح نرم افزار

شما می توانید خطاهای پیش فرض در سطح نرم افزاری را از طریق تغییر در پیکر بندی نرم افزار و یا از طریق افزودن یک Application_Error به فایل Global.asax کنترل نمایید.

شما همچنین می توانید با خطاهای پیش فرض و خطاهای HTTP با افزودن بخش customErrors به فایل web.config برخورد نمایید.بخش customErrors به شما امکان مشخص کردن صفحه ایی را به عنوان صفحه خطا می دهد که در صورت بروز خطا کاربر به آن منتقل می گردد.این بخش همچنین امکان مشخص کردن صفحات جداگانه برای خطاها را در اختیار شما قرار می دهد.

```
<configuration>
<system.web>
<customErrors mode="On"
defaultRedirect="ErrorPage.aspx?handler=customErrors%20section%20-%20Web.config">
<error statusCode="404"
redirect="ErrorPage.aspx?msg=404&amp;handler=customErrors%20section%20-
%20Web.config"/>
</customErrors>
</system.web>
</configuration>
```

متاسفانه،زمانی که شما از این پیکربندی برای انتقال کاربر به صفحه خطا استفاده می نمایید،جزییات خطای رخ داده در دسترس نمی باشد.

اگر چه شما می توانید تمامی خطاهایی که در نرم افزار شما رخ می دهد را از طریق افزودن کد به Application_Error در فایل Global.asax شناسایی کنید و برای کنترل آنها اقدام نمایید.

```
void Application_Error(object sender, EventArgs e)
{
Exception exc = Server.GetLastError();
if (exc is HttpUnhandledException)
{
// Pass the error on to the error page.
Server.Transfer("ErrorPage.aspx?handler=Application_Error%20-%20Global.asax",
true);
}
}
```

کنترل خطا در سطح صفحه

کنترل خطا در این سطح کاربر را به صفحه ایی که خطا در آن رخ داده است بازمی گرداند.به منظور مهیا کردن جزییات خطا ،شما باید جزییات خطا را در صفحه نمایش دهید.

شما ممکن است از کنترل خطا در این سطح به منظور ثبت خطاهایی که کنترل نشده اند و یا هدایت کاربر به صفحه ایی که اطلاعات کمکی را نمایش می دهد استفاده نمایید.

در زیر کدی نمایش داده شده است که تمام خطاهای کنترل نشده را از طریق try/catch دریافت می کند.

```
private void Page_Error(object sender, EventArgs e)
{
Exception exc = Server.GetLastError();
// Handle specific exception.
if (exc is HttpUnhandledException)
{
ErrorMsgTextBox.Text = "An error occurred on this page. Please verify your " +
"information to resolve the issue."
}
// Clear the error from the server.
Server.ClearError();
}
```

بعد از کنترل خطا،شما باید آنرا از طریق فراخوانی متدClearError() از شی Server حذف نمایید،در غیر اینصورت شما خطایی را که قبلا رخ داه است را مشاهده می نمایید.

کنترل خطا در سطح کد

عبارت try/catch شامل یک بلوکtry و چندین بلوک catch می باشد که هر یک نوع خاصی از استثنا را بدام می اندازند.زمانی که یک استثنا اتفاق می افتد CLR بلوک catch مناسب را جستجو می کند.اگر متد جاری یک بلوک catch مناسب را در اختیار نداشته باشدCLR متدی را که متد جاری را فراخوانده است را برای بلوک catch مناسب جستجو می کند و این روند ادامه می یابد.اگر بلوک catch مناسب یافت نگردد CLR یک پیام استثنا کنترل نشده را به کاربر نمایش می دهد و روند اجاری برنامه متوقف می گردد.

کد زیر یک روش معمول برای استفاده از try/catch/finally را برای کنترل خطا نمایش می دهد.

```
try
{
file.ReadBlock(buffer, index, buffer.Length);
}
catch (FileNotFoundException e)
{
Server.Transfer("NoFileErrorPage.aspx", true);
}
catch (System.IO.IOException e)
{
Server.Transfer("IOErrorPage.aspx", true);
}
finally
{
if (file != null)
{
file.Close();
}
}
```

در کد بالا، بلوک try حاوی کدهایی است که باید در مقابل استثنا محافظت گردند.بلوک try تا زمانی که یک استثنا رخ دهد و یا کامل گردد اجرا می گردد.زمانی که یکی از استثناهای FileNotFoundException و یا IOException رخ دهد،اجرا به صفحه دیگری منتقل می گردد.سپس،کدی که در بلوک finally قرار دارد اجرا می گردد.این اجرا در صورت وجود یا عدم وجود خطا صورت می گیرد.

افزودن قابلیت ثبت خطا

در این قسمت شما با نحوه ثبت خطا از طریق افزودن یک کلاس به نام ExceptionUtility آشنا خواهید شد.با انجام این کار ،هر زمان که نرم افزار یک خطا را بدام اندازد ،جزییات خطا در فایل log ثبت می گردد.

۱. یک پوشه به نام Logic به پروژه خود اضافه نمایید.

۲. بر روی پوشه Logic راست کلیک نمایید و عبارت Add->New Item را اتخاب نمایید.پنجره Add New Item نمایش داده می شود.

۳. از سمت چپ عبارت Visual C#->code را انتخاب نمایید.سپس مورد Class را از فهرست میانی انتخاب نمایید و نام ExceptionUtility.cs را برای آن وارد نمایید.

۴. دکمه Add را کلیک نمایید.

۵. . کدهای موجود را با کد زیر جابجا نمایید.

```
public sealed class ExceptionUtility
    {
        private ExceptionUtility()
        { }
        public static void LogException(Exception exc,string source)
        {
            // Include logic for logging exceptions
            // Get the absolute path to the log file
            string logFile = "App_Data/ErrorLog.txt";
            logFile = HttpContext.Current.Server.MapPath(logFile);

            // Open the log file for append and write the log
            StreamWriter sw = new StreamWriter(logFile, true);
            sw.WriteLine("********** {0} **********", DateTime.Now);
            if (exc.InnerException != null)
            {
                sw.Write("Inner Exception Type: ");
                sw.WriteLine(exc.InnerException.GetType().ToString());
                sw.Write("Inner Exception: ");
                sw.WriteLine(exc.InnerException.Message);
                sw.Write("Inner Source: ");
                sw.WriteLine(exc.InnerException.Source);
                if (exc.InnerException.StackTrace != null)
                {
                    sw.WriteLine("Inner Stack Trace: ");
                    sw.WriteLine(exc.InnerException.StackTrace);
                }
            }
            sw.Write("Exception Type: ");
            sw.WriteLine(exc.GetType().ToString());
            sw.WriteLine("Exception: " + exc.Message);
            sw.WriteLine("Source: " + source);
            sw.WriteLine("Stack Trace: ");
            if (exc.StackTrace != null)
            {
                sw.WriteLine(exc.StackTrace);
                sw.WriteLine();
            }
            sw.Close();

        }
    }
```

زمانی که یک استثنا رخ می دهد،استثنا را می توان در یک فایل log و از طریق متدLogException ثبت کرد.این متد دو پارامتر دریافت می کند اولین پارامتر شی Exception می باشد.پارامتر دوم یک string که جزییات منبع استثنا را ثبت می کند.این اطلاعات در فایل ErrorLog.txt در پوشه App_Data ثبت می گردد.

افزودن صفحه خطا

در این پروژه برای نمایش تمام خطاها از یک صفحه استفاده می گردد.این صفحه یک پیام را به کاربر سایت نمایش میدهد.اگر کاربرتوسعه دهنده سایت باشد اطلاعات کامل تر و دقیق تری را نمایش می دهد.

۱.بر روی نام پروژه کلیک راست نمایید و عبارت Add->New Item را انتخاب نمایید.

پنجرهAdd New Item نمایش داده می شود.

۲.از سمت چپ Visual C#->Web را انتخاب نمایید.از فهرست میانی Web Form with Master Page را انتخاب نمایید و نام ErrorPage.aspx را وارد نمایید.

۳.بر روی Add کلیک نمایید.

۴. فایل Site.Master را در صفحه بعد انتخاب نمایید و OK را انتخاب نمایید.

۵.صفحه را به صورت زیر تغییر دهید.

```
<h2>Error:</h2>
<p></p>
<asp:Label ID="FriendlyErrorMsg" runat="server" Text="Label" Font-Size="Large"
style="color: red"></asp:Label>
<asp:Panel ID="DetailedErrorPanel" runat="server" Visible="false">
<p> </p>
<h4>Detailed Error:</h4>
<p>
<asp:Label ID="ErrorDetailedMsg" runat="server" Font-Size="Small" /><br />
</p>
<h4>Error Handler:</h4>
<p>
<asp:Label ID="ErrorHandler" runat="server" Font-Size="Small" /><br />
</p>
<h4>Detailed Error Message:</h4>
<p>
<asp:Label ID="InnerMessage" runat="server" Font-Size="Small" /><br />
</p>
<p>
<asp:Label ID="InnerTrace" runat="server" />
</p>
</asp:Panel>
```

۶. کد پس زمینه(ErrorPage.aspx.cs) را مانند کدهای زیر تغییر دهید.

```
public partial class ErrorPage : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
// Create safe error messages.
string generalErrorMsg = "A problem has occurred on this web site. Please try
again. " +
"If this error continues, please contact support.";
string httpErrorMsg = "An HTTP error occurred. Page Not found. Please try again.";
string unhandledErrorMsg = "The error was unhandled by application code.";
// Display safe error message.
FriendlyErrorMsg.Text = generalErrorMsg;
// Determine where error was handled.
string errorHandler = Request.QueryString["handler"];
```

```
if (errorHandler == null)
{
errorHandler = "Error Page";
}
// Get the last error from the server.
Exception ex = Server.GetLastError();
// Get the error number passed as a querystring value.
string errorMsg = Request.QueryString["msg"];
if (errorMsg == "404")
{
ex = new HttpException(404, httpErrorMsg, ex);
FriendlyErrorMsg.Text = ex.Message;
}
// If the exception no longer exists, create a generic exception.
if (ex == null)
{
ex = new Exception(unhandledErrorMsg);
}
// Show error details to only you (developer). LOCAL ACCESS ONLY.
if (Request.IsLocal)
{
// Detailed Error Message.
ErrorDetailedMsg.Text = ex.Message;
// Show where the error was handled.
ErrorHandler.Text = errorHandler;
// Show local access details.
DetailedErrorPanel.Visible = true;
if (ex.InnerException != null)
{
InnerMessage.Text = ex.GetType().ToString() + "<br/>" +
ex.InnerException.Message;
InnerTrace.Text = ex.InnerException.StackTrace;
}
else
{
InnerMessage.Text = ex.GetType().ToString();

if (ex.StackTrace != null)
{
InnerTrace.Text = ex.StackTrace.ToString().TrimStart();
}
}
}
// Log the exception.
ExceptionUtility.LogException(ex, errorHandler);
// Clear the error from the server.
Server.ClearError();
}
}
```

زمانی که صفحه نمایش خطا نمایش داده می شودرویداد Page_Load اجرا می گردد.در این رویداد اولین مکانی که خطا در آن کنترل شده است معین می گردد.سپس آخرین خطایی که رخ داده است از طریق فراخوانی متدGetLastError شی Server معین می گردد.اگر خطایی وجود نداشته باشد یک خطای کلی ایجاد می گردد.سپس اگر درخواست محلی باشد،تمام جزییات نمایش داده می شوند.بعد از نمایش خطا،خطای رخ داده به فایل Log اضافه می گردد و از سرور حذف می گردد.

نمایش پیام استثنای کنترل نشده در نرم افزار

با افزودن یک بخش customErrors به فایل Web.config شما می توانید خطاهای ساده را کنترل نمایید.شما همچنین می توانید خطاها را بر اساس کد آنها کنترل نمایید به عنوان مثال خطا با کد ۴۰۴ که نشان دهنده عدم یافتن یک فایل است.

بروز رسانی پیکربندی

فایل Web.config را با افزودن یک بخش customErrors بروز نمایید.

۱.فایلWeb.config را در مسیر اصلی پروژه یافته و باز نمایید.

۲.بخش customErrors را به قسمت <system.web> اضافه کنید.

۳.فایل Web.config را ذخیره نمایید.

```
<configuration>
<system.web>
<customErrors mode="On"
defaultRedirect="ErrorPage.aspx?handler=customErrors%20section%20-%20Web.config">
<error statusCode="404"
redirect="ErrorPage.aspx?msg=404&amp;handler=customErrors%20section%20-
%20Web.config"/>
</customErrors>
</system.web>
</configuration>
```

حالت (mode) بخش customErrors در حالت فعال(on) قرار دارد.این بخش همچنین صفحه ایی را که باید در صورت بروز خطا به آن منتقل شد را از طریق خصوصیت defaultRedirect مشخص می کند.در ادامه شما مشخص می نمایید که در صورت بروز خطا با کد ۴۰۴،این خطا چگونه کنترل گردد.

اجرای برنامه

برای مشاهده تغییرات برنامه را اجرا نمایید.

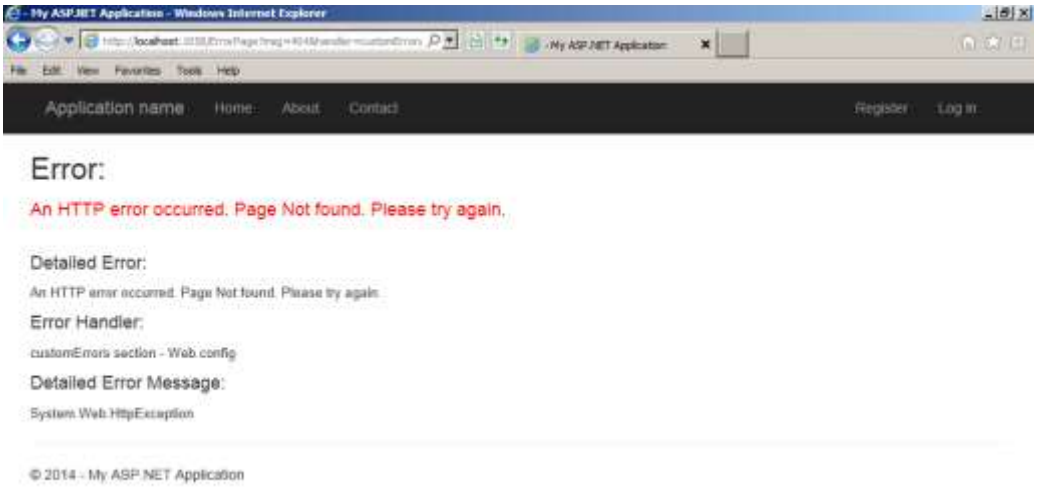
۱.برای اجرا F5 را فشار دهید.

صفحه پیش فرض برنامه نمایش داده خواهد شد.

۲.آدرس زیر را در مرورگر خود وارد نمایید.(شماره پورت شما ممکن است متفاوت باشد.):

http://localhost:1058/NoPage.aspx

۳.صفحه ErrorPage.aspx نمایش داده می شود.



افزودن یک خطای عمدی

به منظور ارزیابی نحوه عملکرد برنامه در زمان مواجه با خطا ،می توان یک خطای عمدی را ایجاد نمود.ما این خطای عمدی را در صفحه نخست برنامه ایجاد می کنیم.

۱.کد پس زمینه صفحه Default.aspx را باز نمایید.

۲.کد زیر را به بخش Page_Load اضافه نمایید.

```
protected void Page_Load(object sender, EventArgs e)
{
throw new InvalidOperationException("An InvalidOperationException " +
"occurred in the Page_Load handler on the Default.aspx page.");
}
```

می توان استثناهای گوناگونی را ایجاد نمود.در کد بالا یک نمونه از استثنا InvalidOperationException ایجاد شده است.

اجرای برنامه

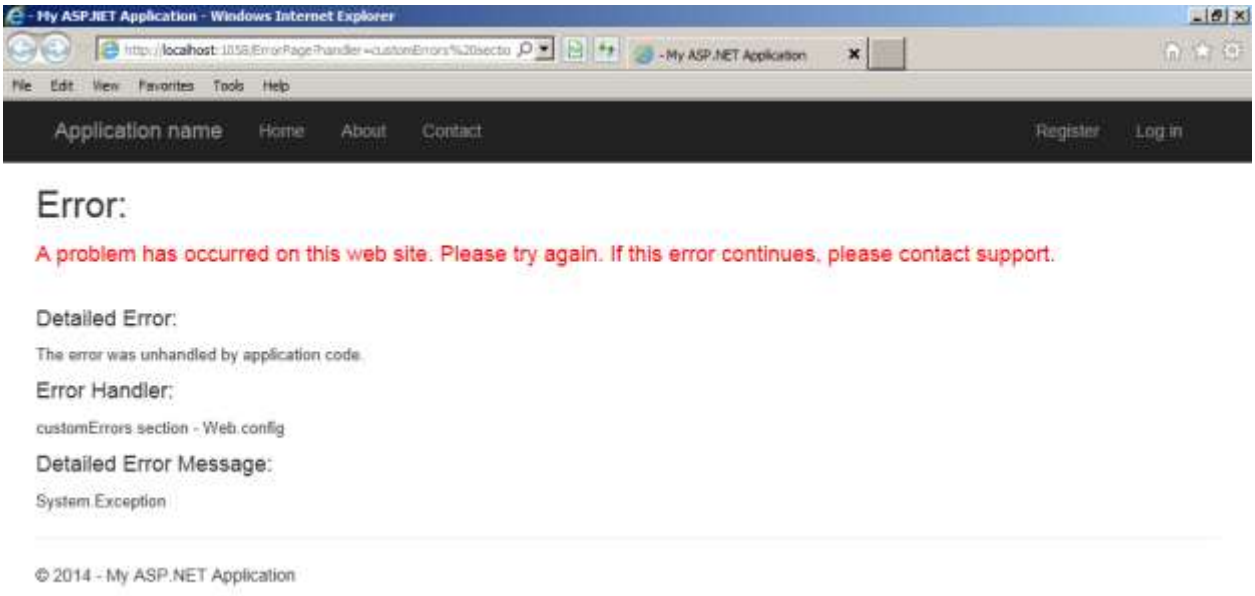
برای نحوه عملکرد برنامه آنرا اجرا نمایید.

۱.برای اجرا CTRL+F5 را فشار دهید.

استثنا InvalidOperationException ایجاد خواهد شد.

تذکر: به منظور اجرای برنامه در حالتی که در صورت بروز خطا کد برنامه نمایش داده نشود باید از CTRL+F5 استفاده نمایید.

۲.صفحه ErrorPage.aspx به همراه اطلاعات خطا نمایش داده می شود.



افزودن کنترل خطا در سطح نرم افزار

به جای استفاده از customErrors در فایل Web.config که اطلاعات مختصری در مورد خطاها را در دسترس قرار می دهد شما می توانید خطاها را در سطح نرم افزار کنترل نمایید.

۱.در Solution Explorer فایل Global.asax را یافته و باز نمایید.

۲.یک رسیدگی کنندهApplication_Error را به همراه کدهای زیر اضافه نمایید.

```
void Application_Error(object sender, EventArgs e)
{
// Code that runs when an unhandled error occurs.
// Get last error from the server
Exception exc = Server.GetLastError();
if (exc is HttpUnhandledException)
{
if (exc.InnerException != null)
{
exc = new Exception(exc.InnerException.Message);

Server.Transfer("ErrorPage.aspx?handler=Application_Error%20-%20Global.asax",
true);
}
}
}
```

زمانی که یک خطا رخ می دهد، متد Application_Error فراخوانی می گردد.در این متد آخرین استثنا بازیابی می گردد.اگر استثنا کنترل نشده باشد و یا دارای یک استثنا داخلی باشد،نرم افزار اجرا را به صفحه ایی که جزییات استثنا را نمایش می دهد منتقل می کند.

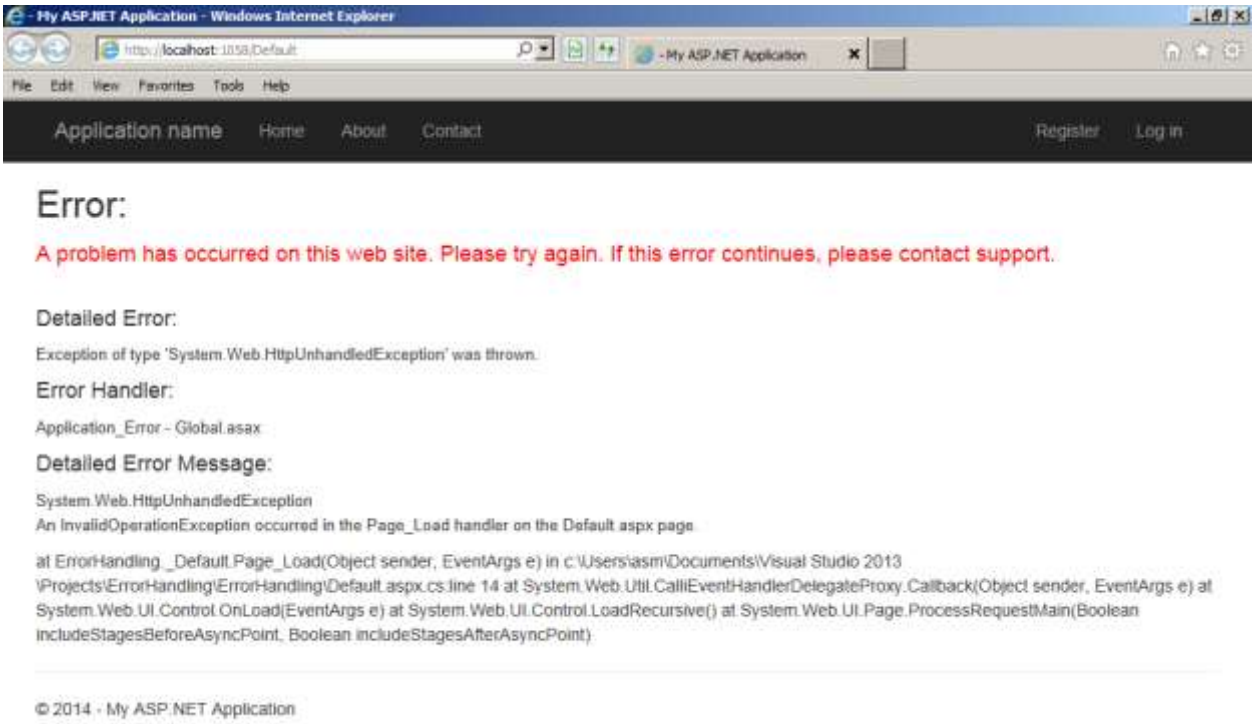
اجرای برنامه

شما می توانید تغییرات اعمال شده را با اجرای برنامه مشاهده نمایید.

۱.برای اجرای برنامه CTRL+F5 را فشار دهید.

استثنای InvalidOperationException در صفحه Default.aspx که در مرحله قبل آن را ایجاد نمودیم رخ می دهد.

۲.صفحه ErrorPage.aspx نمایش داده می شود.



افزودن کنترل خطا در سطح صفحه

شما می توانید خطا را در سطح صفحه از طریق افزودن خصیصه ErrorPage به صفحه و یا با افزودن یک رسیدگی کننده Page_Error به کدهای پس زمینه یک صفحه کنترل نمایید.در این قسمت،شما یک رسیدگی کننده Page_Error را به صفحه اضافه خواهید کرد که این کد اجرا را به صفحه ErrorPage.aspx منتقل می کند.

۱.در Solution Explorer فایل Default.aspx.cs را یافته و باز نمایید.

۲.یک رسیدگی کنندهPage_Error را به کد اضافه نمایید درنتیجه صفحه مانند زیر تغییر خواهد کرد.

```
public partial class _Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            throw new InvalidOperationException("An InvalidOperationException " +
"occurred in the Page_Load handler on the Default.aspx page.");
        }
        private void Page_Error(object sender,EventArgs e)
        {
            //get last error from the server
            Exception exc = Server.GetLastError();
            //Handle specific exception
            if(exc is InvalidOperationException)
            {
                //pass the error on to error page
                Server.Transfer("ErrorPage.aspx?handler=Page_Error%20-
%20Default.aspx", true);
            }
        }

    }
```

زمانی که یک خطا رخ می دهد،Page_Error فراخوانی می گردد.در این رسیدگی کننده آخرین استثنا بازیابی می گردد.اگر InvalidOperationException اتفاق افتاده باشد،این رسیدگی کننده اجرا را به صفحهErrorPage.aspx منتقل می کند.

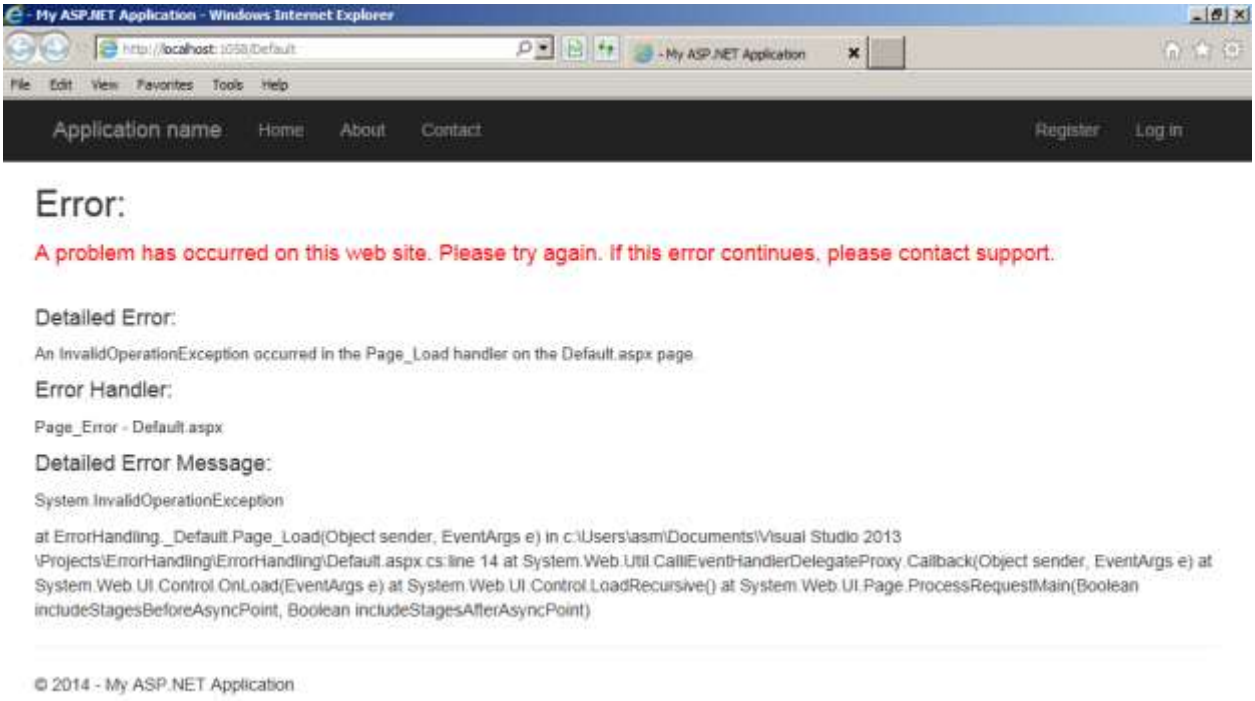
اجرای برنامه

شما می توانید تغییرات اعمال شده را با اجرای برنامه مشاهده نمایید.

۱. .برای اجرای برنامه CTRL+F5 را فشار دهید.

استثنای InvalidOperationException در صفحه Default.aspx که در مرحله قبل آن را ایجاد نمودیم رخ می دهد.

۲.صفحه ErrorPage.aspx نمایش داده می شود.



افزودن کنترل خطا در سطح کد

همان طور که قبلا ذکر شد،شما با فزودن عبارات try/catch می توانید خطاهایی اولیه که رخ می دهند را کنترل نمایید.در این مثال شما تنها اطلاعات خطا را در فایل log ثبت خواهید کرد که این اطلاعات می توانند بعدا بازبینی شوند.

۱.کد پس زمینه صفحهDefault.aspx را به صورت زیر تغییر دهید.

```
public partial class _Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                throw new InvalidOperationException("An InvalidOperationException " +
"occurred in the Page_Load handler on the Default.aspx page.");
            }
            catch(Exception exc)
            {
                ExceptionUtility.LogException(exc, "Page_Load in Default.aspx");
            }

        }
        private void Page_Error(object sender,EventArgs e)
        {
            //get last error from the server
            Exception exc = Server.GetLastError();
            //Handle specific exception
            if(exc is InvalidOperationException)
            {
                //pass the error on to error page
                Server.Transfer("ErrorPage.aspx?handler=Page_Error%20-
%20Default.aspx", true);
            }
        }

    }
```

کد بالا اطلاعات مربوط به خطا را در فایل log ذخیره می کند.

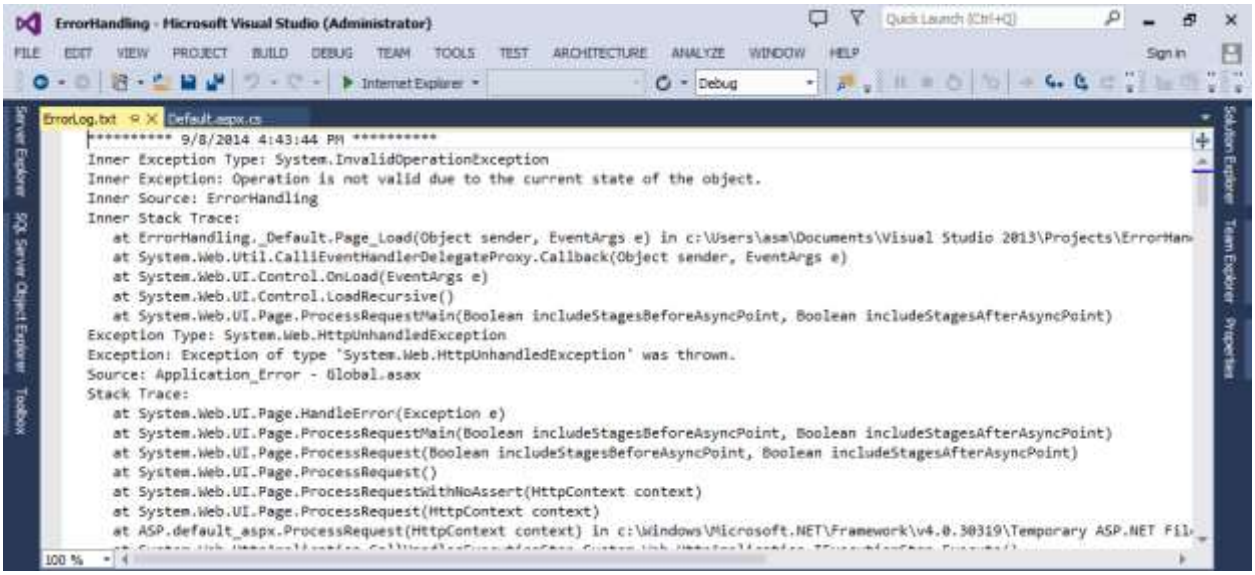
بررسی اطلاعات خطاهای ثبت شده

همان طور که قبلا ذکر شد،شما می توانید از فایل ErrorLog.txt برای معین کردن اینکه چه خطاهایی رخ داده اند و باید ابتدا رسیدگی شوند استفاده نمایید.اگر چه تنها خطاهایی که بدام افتاده اند و در این فایل نوشته شده اند گزارش می شوند.

۱.در Solution Explorer فایل ErrorLog.txt را یافته و باز نمایید.

برای نمایش این فایل ممکن است نیاز باشد تا شما دکمه "Show All Files" را در Solution Explorer انتخاب نمایید.

۲.خطاهای ثبت شده در این فایل را مشاهده نمایید.



پیام های خطا

توجه به این نکته نیز ضروری است که در هنگام نمایش پیام خطا اطلاعات حساس را در اختیار کاربران قرار ندهید.به عنوان مثال،اگر برنامه شما در نوشتن اطلاعات در پایگاه داده ناموفق است ،نیازی نیست کاربر از این موضوع مطلع گردد و یا پیام به گونه ایی نمایش داده شود که حاوی نام کاربری پایگاه داده باشد.