Elmah Everywhere Documentation

Elmah Everywhere is an exception logging library for .NET, Silverlight, WPF, ASP.NET MVC and WCF that uses an ELMAH (Error Logging Modules and Handlers for ASP.NET).

For more detailed information how to configure and use Elmah Everywhere see source code and samples at <https://github.com/vincoss/vinco-logging-toolkit>.

# Error website default login details

* UserName : administrator
* Password: p@ssword

# Silverlight configuration

To configure Silverlight exception logging you must add a reference to Elmah.Everywhere.Silverlight.dll and configure exception handler.

private static void SetUpExceptionHandler()

{

// Configure

var writter = new ClientHttpExceptionWritter

{

// **NOTE: Possible to pass URI by startup arguments.**

RequestUri = new Uri("http://localhost:11079/error/log", UriKind.Absolute)

};

Uri uri = Application.Current.Host.Source;

var defaults = new ExceptionDefaults

{

Token = "Silverlight-Test-Token",

ApplicationName = "Silverlight-Sample",

Host = string.Format("{0}{1}{2}:{3}", uri.Scheme, Uri.SchemeDelimiter, uri.Host, uri.Port)

};

ExceptionHandler.Configure(writter, defaults);

}

**In Application constructor call handler setup method.**

public App()

{

SetUpExceptionHandler();

}

**Add handler log into Application\_UnhandledException method**

private void Application\_UnhandledException(object sender, ApplicationUnhandledExceptionEventArgs e)

{

ExceptionHandler.Report(e.ExceptionObject, null);

}

# WPF configuration

To configure WPF exception logging you must add a reference to Elmah.Everywhere.dll and configure exception handler.

**In Application constructor call handler setup method.**

public App()

{

// Configure error handler from configuration file

ExceptionHandler.ConfigureFromConfigurationFile(new HttpExceptionWritter());

}

**Attach handler.**

protected override void OnStartup(StartupEventArgs e)

{

ExceptionHandler.Attach(AppDomain.CurrentDomain);

}

**Detach handler.**

protected override void OnExit(ExitEventArgs e)

{

ExceptionHandler.Detach(AppDomain.CurrentDomain);

}

**Configuration details in App.config file**

<everywhere>

<!-- Configure Elmah.Everywhere

URL: Remote web site url to log an error

Token: Token to identify client

ApplicationName: Error source

Host: Error host

-->

<settings remoteLogUri="http://localhost:11079/error/log"

token="Token-Test"

host="Wpf-Sample"

applicationName="Exceptions-Handler"/>

</everywhere>

# Code configuration

To configure exception logic from code you must add a reference to Elmah.Everywhere.Silverlight.dll and configure exception handler.

**Configure exception handler**

static void Main(string[] args)

{

var writter = new HttpExceptionWritter

{

RequestUri = new Uri("http://localhost:11079/error/log", UriKind.Absolute)

};

var defaults = new ExceptionDefaults

{

Token = "Test-Token",

ApplicationName = "Console-Sample",

Host = Environment.MachineName

};

ExceptionHandler.Configure(writter, defaults);

}

# WCF configuration with config file

To configure WCF exception logging you must add those settings into web.config file.

<system.serviceModel>

<serviceHostingEnvironment aspNetCompatibilityEnabled="true" multipleSiteBindingsEnabled="true" />

<extensions>

<behaviorExtensions>

<add name="elmah" type="Elmah.Everywhere.ServiceModel.ErrorBehaviorExtensionElement, Elmah.Everywhere"/>

</behaviorExtensions>

</extensions>

<behaviors>

<serviceBehaviors>

<behavior>

<elmah/>

</behavior>

</serviceBehaviors>

</behaviors>

</system.serviceModel>

# WCF configuration with behaviour attribute

Decorate your WCF or WCF RIA services with ServiceHttpErrorBehaviour attribute.

[ServiceHttpErrorBehavior(typeof(HttpErrorHandler))]

public class UserRegistrationService : DomainService

{

}

# Trace diagnostics configuration

Configure trace to log errors into a file. Trace log can be also used to diagnose issues with Elmah.Everywhere error log.

<configuration>

<system.diagnostics>

<trace autoflush="true">

<listeners>

<clear/>

<add name="fileListener" type="System.Diagnostics.TextWriterTraceListener" initializeData="E:\\_Log\Console\_Sample\_FileListener.log" />

</listeners>

</trace>

</system.diagnostics>

</configuration>