Savonia.Web.ErrorReporter.dll usage

1. Make a reference to Savonia.Web.ErrorReporter.dll
2. Set settings to web.config

To send error mails add value for ExceptionMailsRecipients in appSettings and set mailSettings:

Note 1: With default settings posti.savonia.fi won’t relay emails outside Savonia!

<appSettings>

<add key="ExceptionMailsRecipients" value="[recipients as comma separated list goes here]@savonia.fi" />

</appSettings>

<system.net>

<mailSettings>

   <smtp deliveryMethod="Network" from="[sender goes here]@savonia.fi">

     <network host="posti.savonia.fi" port="25" defaultCredentials="true" />

    </smtp>

  </mailSettings>

</system.net>

Note 2: If ExceptionMailsRecipients is not defined nor has value, then the exception is written to Trace!

1. Plug-in error reporter module in web.config:

<system.web>

    <httpModules>

      <add type="Savonia.Web.ErrorReporter.ErrorReporterModule" name="ErrorReporterModule" />

     </httpModules>

</system.web>

   <system.webServer>

      <modules runAllManagedModulesForAllRequests="true">

       <add type="Savonia.Web.ErrorReporter.ErrorReporterModule" name="ErrorReporterModule" />

     </modules>

<system.webServer>

Asp.Net MVC error handling in Global.asax

protected void Application\_Error()

{

#if DEBUG

return;

#endif

// <http://www.west-wind.com/weblog/posts/2009/Apr/29/IIS-7-Error-Pages-taking-over-500-Errors>

// try to skip IIS 7 error messages.

Response.TrySkipIisCustomErrors = true;

// <http://stackoverflow.com/questions/5226791/custom-error-pages-on-asp-net-mvc3>

var exception = Server.GetLastError();

var httpException = exception as HttpException;

Response.Clear();

Server.ClearError();

var routeData = new RouteData();

routeData.Values["controller"] = "Error";

routeData.Values["action"] = "Index";

routeData.Values["exception"] = exception;

Response.StatusCode = 500;

if (httpException != null)

{

Response.StatusCode = httpException.GetHttpCode();

switch (Response.StatusCode)

{

case 403:

routeData.Values["action"] = "NotAuthorized";

break;

case 404:

routeData.Values["action"] = "NotFound";

break;

}

}

IController errorController = new ErrorController();

var rc = new RequestContext(new HttpContextWrapper(Context), routeData);

errorController.Execute(rc);

}