

Re: HttpModules – Webservices

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-05/msg02107.html>

- *From:* Scott Allen <scott@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 May 2005 16:52:59 -0400
-

Bruce:

The Application.Lock would make me nervous. It does look as if you have an Unlock at the right places, but you might consider using a finally clause. I'd

When the application blocks can you break in with the debugger and look at call stacks for all the threads? That might show you how the application became deadlocked.

—

Scott

<http://www.OdeToCode.com/blogs/scott/>

On Wed, 11 May 2005 20:30:57 +0100, "Bruce Hodge"
<support@xxxxxxxxxxxxxxxxxxxxxx> wrote:

>Hi,
>
>We have a couple of clients who are having trouble connecting to .Net
>webservices, (they're coming from PHP and VFP). We were looking to provide
>a diagnostic tool that displayed the soap message submitted so that if the
>call failed they could compare their SoapMessage with the ones we generate
>in .Net.
>
>Thought the solution was an HttpModule hooked into begin request, but after
>approx 10 requests it hangs the webservice/aspnet service, any thoughts?
>Code below.
>
>(The HttpRequest are stored in Typed Dataset, stored on the Application
>object so that they can be accessed from webpages.)
>
>
>public void OnBeginRequest(Object s, EventArgs e)
>{
> try
> {

Re: HttpModules – Webservices

```
> // Cast the Object to the HttpApplication variable.
> HttpApplication app = (HttpApplication)s;
>
> HttpRequest newRequest = app.Request;
>
> string servers = this.ServicesToMonitor();
> int posn =
> servers.ToLower().IndexOf(this.GetServiceName(newRequest).ToLower());
>
> if (posn > -1)
> this.AddFromHttpRequest(this.RequestHistory, newRequest);
>
> }
>
> catch (Exception ex)
> {
> //Custom error logging code
> }
>
>}
>
>private void AddFromHttpRequest(Mars.Schema.HttpRequests history,
>HttpRequest request)
>{
> try
> {
> HttpContext.Current.Application.Lock();
>
> byte [] byteArray = new byte[request.ContentLength];
>
> request.InputStream.Read(byteArray, 0, request.ContentLength);
>
> // Must reset the Input Stream position or the whole process fails.
> request.InputStream.Position = 0;
>
> Mars.Schema.HttpRequests.RequestsRow row =
> this.RequestHistory.Requests.AddRequestsRow(
>
>
> request.RawUrl,
> request.HttpMethod,
> request.ContentType,
> request.ContentLength,
> Encoding.ASCII.GetString(byteArray, 0, request.ContentLength),
> DateTime.Now,
> this.GetServiceName(request),
> this.SOAPAction(request),
> request.ServerVariables["REMOTE_ADDR"]);
>
> foreach (string s in request.Headers.AllKeys)
> this.RequestHistory.HttpHeaders.AddHttpHeadersRow(row.RequestID,
```

