ALLAN THRAEN | 16 years ago | PDF |

NIGHTLY FUN WITH 301

WARNING: GEEKY STUFF.

Do you ever have difficulty falling asleep at night, because your brain begins to code the moment your tired body hits the bed? Well, I do every once in a while. And then I know that I can either toss and turn all night or sneak downstairs and code for a while - just to get it out of my system.

Last night was one of those nights. This time, though, I decided to make something simple but useful. The idea had been in the back of my head ever since I visited a large client recently who was getting ready to move a big and popular web site in the media industry to EPiServer CMS 5.

Basically this is the classic problem that I'm trying to solve: When moving a web site to a new platform it often happens that the url to the individual pages changes which breaks links, search engine ranks and results in a lot of 404 errors. For instance, with Friendly URLS in EPiServer a page that used to be called "/articles/article124145.html" might now get the friendlier path"/News/New+Website+Launched/". So, my solution to this problem is simply a generic HttpModule that loads a series of old-path / new-path sets from a text-file when the web application is started and then makes a quick check for every request to see if it's actually an old URL being requested. If it is, it'll do the correct thing and send back an HTTP 301 Permanently Moved reply with the new location of the document. That way both global search engines and various caches should get updated and no links will be broken. Simple, but it seems to work - and to my luck it was so easy to code that I still got most of a good nights sleep.

```
public class RedirectModule : IHttpModule
{
    private Dictionary<string, string> map;

    public void Init(HttpApplication app)
    {
        app.BeginRequest += new EventHandler(app_BeginRequest);
        map = new Dictionary<string, string>();
        //Load from file
        StreamReader sr = File.OpenText(ConfigurationManager.AppSettings["UrlMapping"]);
        string s = null;
        while ((s = sr.ReadLine()) != null)
        {
            string[] parts = s.Split('|');
            if(parts.Length==2)  map.Add(parts[0], parts[1]);
        }
        sr.Close();
    }

    void app_BeginRequest(object sender, EventArgs e)
    {
        HttpApplication app = (sender as HttpApplication);
        if (map.ContainsKey(app.Request.Url.LocalPath))
        {
            app.Response.Status = "301 Moved Permanently";
            app.Response.AddHeader("Location", map[app.Request.Url.LocalPath]);
            app.Response.End();
        }
    }

    public void Dispose()
    {
    }
}
```

As said - it didn't keep me up all night.

In order for it to work the following adjustments needs to go into web.config:

```
<appSettings>
 <add key="UrlMapping" value="c:\\inetpub\\UrlMapping.txt"/>
</appSettings>

<system.web>
 <httpModules>
  <add name="RedirectModule"
type="EPiServer.Research.RedirectModule.RedirectModule,EPiServer.Research.RedirectModule"/>
 </httpModules>
</system.web>
```

And the mapping file should just be a series of [path]|[new path], one on each line like this:

```
/oldpath/article.html|/newpath/newarticle.aspx
/oldpath/oldpage|http://www.google.com/?q=oldpage+topic
```

CodeArt ApS
Teknikerbyen 5, 2830 Virum, Denmark
Email: info@codeart.dk
Phone: +45 26 13 66 96
CVR: 39680688