ASP.NET MVC - Routing of Legacy URLs

When you convert an existing ASP.NET application to ASP.NET MVC you will sooner or later want to handle legacy URLs.

Imagine an ASP.NET application containing the page "YourPage.aspx".
Now you want to redirect request to that legacy URL to the new MVC Action: "Home/YourPage".

I have looked at several implementations on the internet, but none of them was really simple.
Thus I implemented another solution which is really easy to configure.

For the URL "YourPage.aspx" (see above) you only have to add one line to your Global.asax.cs:

```
routes.Add(new LegacyRoute("YourPage.aspx", "Home/YourPage"));
```

Quite simple, isn't it? But what happens behind the scenes? Let's have a look at the source:

The class `LegacyRoute` derives from `Route` and its constructor takes a second argument "target", which will be the RedirectLocation:

```
public class LegacyRoute : Route
{
    public LegacyRoute(string url, string target)
        : base(url, new LegacyRouteHandler())
    {
        this.Target = target;
    }

    public string Target { get; private set; }

    public override VirtualPathData GetVirtualPath(RequestContext
requestContext, RouteValueDictionary values)
    {
        return null;
    }
}
```

The constructor in `LegacyRoute` uses `LegacyRouteHandler`, this class simply creates instances of `LegacyHandler`:

```
public class LegacyRouteHandler : IRouteHandler
{
    public IHttpHandler GetHttpHandler(RequestContext requestContext)
    {
        return new LegacyHandler(requestContext);
    }
}
```

And finally the `LegacyHandler` which is responsible for the redirect:

```
public class LegacyHandler : MvcHandler
{
    public LegacyHandler(RequestContext requestContext)
        : base(requestContext)
    {
    }

    protected override void ProcessRequest(HttpContextBase httpContext)
    {
        var legacyRoute = RequestContext.RouteData.Route as LegacyRoute;

        httpContext.Response.Status = "301 Moved Permanently";
        httpContext.Response.RedirectLocation = legacyRoute.Target;
    }
}
```

Now if you call "YourPage.aspx" the `LegacyRoute` matches the request. Instead of the regular `MVCHandler` a `LegacyHandler` redirects the request to "Home/YourPage".

I have attached the full source code including a demo website so you see the above code in action.