

Cache-control and Expires headers

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2008-05/msg00165.html>

- *From:* Mike Kraley <mkraley@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 2 May 2008 14:52:02 -0700
-

In my application, I'm using a wildcard mapping so that all requests go via ASP.NET. Then I'm using an `HttpModule` to look at all the requests and do some URL rewriting. Some of the requests then also pass thru custom `HttpHandlers`, based on extension.

For some file types, e.g. `.js` and `.swf`, I want these files to be cached on the browser. I am trying to set `Cache-Control` to `public`, and would like either no `Expires` header, or at least something with a long expiration date, e.g. 1 month or longer.

You may ask why I'm doing all this: We'd like our `.swf` and `.js` files to be cached in the user's browser – they are big, so we'd like them to stay around. When we update our code, we will effectively change our URL, which will cause a new version to be downloaded. But we serve our files under SSL, and firefox 3, at least, seems to require an explicit `cache-control: public` in order to cache SSL content.

So I've tried setting the various properties of `Response.Cache` in different event handlers of the `HttpModule`. Setting `cacheability` to `Public` seems to work fine, but then the response comes out with an expiration date of 1 day in the future. I then tried to use `cache.SetExpires` to lengthen the expiration date. Doesn't work.

A lot of reading of reflector later, what I learned is that something in the code I can't see is setting `_utcExpires` to 1 day after now. But the logic in the `HttpCachePolicy` object uses the minimum value of any call to `SetExpires`. So if I set the expiration date to 1 month early in the pipeline, e.g. at `BeginRequest` time, the value is subsequently overridden to 1 day. If I try to overwrite the value late in the pipeline (anything after `PostRequestHandlerExecute`) nothing happens since the headers have already been written.

[In IIS5, I used a hack that used reflection to set the private variable `_isExpiresSet` to `false`; then I could set a new value with `SetExpires`. But in IIS6, which is where I have to get this to work, that doesn't work since the headers have been written.]

I tried setting expiration in the IIS management tool, but that just affects

Cache-control and Expires headers

the max-age value in Cache-control, but doesn't include public.

Any thoughts on how I can get IIS to vend the headers I want?

--

....Mike

.