

قسمت‌های اول تا سوم این مقاله: [+](#) و [+](#) و [+](#)

در قسمت چهارم قصد داریم هدر مربوط به Content Expiration Date را توسط یک Http module به محتوای غیرپویای سایت مانند تصاویر ، فایل‌های CSS و غیره اعمال کنیم. این روش از روش قسمت دوم ساده‌تر است و جامع‌تر. ابتدا یک پروژه‌ی Class library جدید را به نام StaticContentCacheModule ایجاد کرده و سپس ارجاعی را به اسمبلی استاندارد System.Web.dll به آن خواهیم افزود. سپس کدهای مرتبط با این ماژول به شرح زیر هستند:

```
//StaticCache .cs
using System;
using System.Web;

namespace StaticContentCacheModule
{
    public class StaticCache : IHttpModule
    {
        public void Init(HttpApplication context)
        {
            context.PreSendRequestHeaders += context_PreSendRequestHeaders;
        }

        static void context_PreSendRequestHeaders(object sender, EventArgs e)
        {
            //capture the current Response
            var currentResponse = ((HttpApplication)sender).Response;

            if (CacheManager.ShouldCache(currentResponse.ContentType))
            {
                currentResponse.AddHeader("cache-control", "public");
                currentResponse.AddHeader("Expires",
                    DateTime.Now.Add(TimeSpan.FromDays(30)).ToString());
            }
        }

        public void Dispose() { }
    }
}
```

در اینجا ContentType تک تک عناصری که توسط وب سرور ارائه خواهند شد، بررسی می‌شود. اگر نیازی به کش شدن آن‌ها وجود داشت (توسط کلاس CacheManager این امر مشخص می‌گردد)، هدر مربوطه اضافه می‌گردد.

```
//CacheManager.cs
using System;

namespace StaticContentCacheModule
{
    class CacheManager
    {
        public static bool ShouldCache(string contentType)
        {
            contentType = contentType.ToLower();
            string[] parts =
                contentType.Split(
                    new[] { ';' },
                    StringSplitOptions.RemoveEmptyEntries
                );

            if (parts.Length > 0)
                contentType = parts[0];

            bool cache = false;
        }
    }
}
```

```

switch (contentType)
{
    case "text/css":
        cache = true; break;
    case "text/javascript":
    case "text/jscript":
        cache = true; break;
    case "image/jpeg":
        cache = true; break;
    case "image/gif":
        cache = true; break;
    case "application/octet-stream":
        cache = true; break;
    default:
        {
            if (contentType.Contains("javascript"))
                cache = true;
            if (contentType.Contains("css"))
                cache = true;
            if (contentType.Contains("image"))
                cache = true;
            if (contentType.Contains("application"))
                cache = true;
        }
        break;
}

return cache;
}
}
}

```

در این کلاس، contentType دریافتی بررسی می‌شود. اگر نوع محتوای قابل ارائه از نوع JavaScript، CSS، تصویر و یا Application بود، یک مقدار true بازگشت داده خواهد شد. نهایتاً برای استفاده از این Http module جدید در IIS6 به قبل در وب کانفیگ برنامه خواهیم داشت:

```

<httpModules>
  <add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache,
StaticContentCacheModule"/>
</httpModules>

```

و یا در IIS7 این تغییرات به صورت زیر می‌تواند باشد:

```

<system.webServer>
  <modules>
    <add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache,
StaticContentCacheModule"/>
  </modules>

```

اکنون اگر یک پروژه‌ی آزمایشی جدید ASP.Net را گشوده و فایل css ساده‌ای را به آن اضافه کنیم، بررسی هدر نهایی توسط افزونه‌ی YSlow به صورت زیر خواهد بود:

URL	EXPIRES (Y/M/D)
http://localhost/TestStaticContentCacheModule/	no expires
http://localhost/TestStaticContentCacheModule/Stylesheet1.css	2010/2/10

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۰۹:۴۴:۱۴

نکته: اگر کلاس‌های فوق را به برنامه‌ی خود اضافه کرده‌اید و ماژول‌ها کار نمی‌کنند باید به این مطلب دقت داشته باشید: [\(+\)](#)

نویسنده: mohammad
تاریخ: ۱۳۸۹/۱۰/۲۰ ۱۹:۲۵:۱۰

سلام آقا بازم نمی‌شه من کد رو به vb تبدیل کردم و در وب کانفیگ هم به صورت زیر گذاشتم:

```
add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache, App_Code.7pqudgp-,  
"Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
```

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۲۲:۱۳:۳۳

سورس کامل مرتبط با این مطلب در همان زمان: [\(+\)](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۲۳:۴۳:۲۳

اگر روش فوق برای شما کار نمی‌کند، روش ذکر شده در این مطلب: [\(+\)](#) حتما کار خواهد کرد و حداقل برای اسکریپت‌های سایت می‌توانید از آن استفاده کنید.

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۰:۴۰:۲۵

بنظرم، مشکل با عبارت This operation requires IIS integrated pipeline mode ارتباط دارد که هم در IIS6 و IIS7 وجود دارد

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۱:۳۲:۳۱

فقط IIS های 7 به بعد integrated pipeline mode را دارند. به همین جهت بجای `currentResponse.Headers[something]` از `currentResponse.AddHeader` استفاده شده.

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۹/۱۰/۲۲ ۱۳:۳۷:۳۴

من مشکل رو در iis7 در ویندوز 7 مبینم و نکته دیگه اینکه در ii6 و ویندوز سرور 2003 هم کار نکرده و با اینکه این چند روز خیلی دنبال این مسئله تو اینترنت بودم حدود 70 مقاله که اکثرا شبیه به هم بودند رو بررسی کردم حتی iis7 رو دوباره نصب کردم ولی مشکل همچنان باقیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۲ ۱۳:۵۵:۳۰

اگر دسترسی به IIS دارید که نیازی به این مقاله نیست. باید به اینجا مراجعه کنید: [\(+\)](#)
در مورد IIS-7 هم اطلاعات بیشتر اینجا است: [\(+\)](#)

در IIS7 این مورد توسط انتخاب application pool مناسب تنظیم می‌شود. برای مثال Classic mode آن با IIS6 سازگار است.