MEZIANTOU'S BLOG

Blog about Microsoft technologies (.NET, .NET Core, ASP.NET Core, WPF, UWP, TypeScript, etc.)

HOME PROJECTS ARCHIVES CONTACT

Caching static resources forever with ASP.NET Core

01/08/2017 Gérald Barré .NET, ASP.NET Core, Web

Caching static resources like stylesheet, JavaScript, or image files allows improving the performance of your website. On the client-side, a static file will always be loaded from the cache which reduces the number of requests to the server and so, the time to get the page and its resources. On the server-side, as they are fewer requests, the server can handle more clients without upgrading the hardware.

While caching is a great thing, you must ensure the client is always running the latest version of the application. I mean, when you deploy the next version of the website, you don't want the client to use an obsolete cached version of a file.

Versioned URL (Cache buster)

To ensure the user always uses the latest version of a file, we must have a unique URL per version of a file. There are many strategies:

- Use the query string: `https://sample.com/file.js?v=123`
- Rename the file: `https://sample.com/file.123.js`
- Create a directory: `https://sample.com/123/file.js`

ASP.NET Core provides a mechanism using a `TagHelper` to append the version with the query string. It supports the most common HTML tags that target a static resource: `script` , `link` and `img` . All you have to do is append `asp-append-version="true"` to the tag:

```
<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
<script src="~/js/site.js" asp-append-version="true"></script>
<img src="~/images/banner1.svg" asp-append-version="true" />
```

This will produce:

GÉRALD BARRÉ

aka. meziantou

```
<link rel="stylesheet" href="/css/site.css?v=1wp5zz4e-mOPFx4X2O8seW_DmUtePn5xFJk1vB7JKRc" />
<script src="/js/site.js?v=EWaMeWsJBYWmL2g_KkgXZQ5nPe-a3Ichp0LEgzXczKo"></script>
<img src="/images/banner1.svg?v=GaE_EmkeBf-yBbrJ26lpkGd4jkOSh1eVKJaNOw9I4uk" />
```

The version is the SHA256 of the file encoded in base64. Of course, the hash is computed only once per file and stored in an `IMemoryCache` .

The URL of files are now unique and will change when the file change, so we can add a cache header to the response to indicate the client that the file can be stored in cache forever.

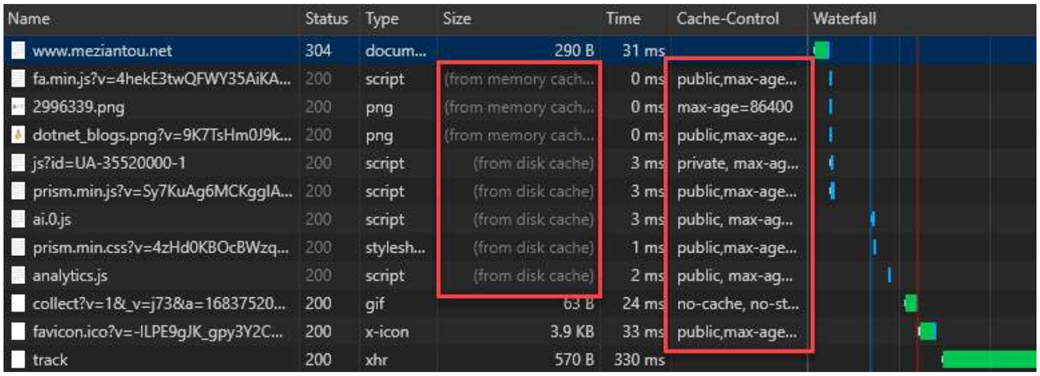
Response headers

To indicate the browser to store the file in its cache, we must send the `Cache-control` header and the `Expires` header for `HTTP/1.0` compatibility. To add these headers, we use the `OnPrepareResponse` callback of the `StaticFilesOptions` . Let's modify the `Startup.cs` file:

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseStaticFiles(new StaticFileOptions
    {
        OnPrepareResponse = context =>
        {
            // Cache static file for 1 year
            if (!string.IsNullOrEmpty(context.Context.Request.Query["v"]))
            {
                context.Context.Response.Headers.Add("cache-control", new[] { "public,max-age=3
                context.Context.Response.Headers.Add("Expires", new[] { DateTime.UtcNow.AddYear
            }
        }
    });

    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "default",
            template: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

You can check the headers are sent by using the Developer Console:



Name	Status	Type	Size	Time	Cache-Control	Waterfall
www.meziantou.net	304	docum...	290 B	31 ms		
fa.min.js?v=4hekE3twQFWY35AiKA...	200	script	(from memory cach...	0 ms	public,max-age...	
2996339.png	200	png	(from memory cach...	0 ms	max-age=86400	
dotnet_blogs.png?v=9K7TsHm0J9k...	200	png	(from memory cach...	0 ms	public,max-age...	
js?id=UA-35520000-1	200	script	(from disk cache)	3 ms	private, max-ag...	
prism.min.js?v=Sy7KuAg6MCKggIA...	200	script	(from disk cache)	3 ms	public,max-age...	
ai.0.js	200	script	(from disk cache)	3 ms	public, max-ag...	
prism.min.css?v=4zHd0KBOcBWzq...	200	stylesh...	(from disk cache)	1 ms	public,max-age...	
analytics.js	200	script	(from disk cache)	2 ms	public, max-ag...	
collect?v=1&_v=j73&a=16837520...	200	gif	63 B	24 ms	no-cache, no-st...	
favicon.ico?v=-ILPE9gJK_gpy3Y2C...	200	x-icon	3.9 KB	33 ms	public,max-age...	
track	200	xhr	570 B	330 ms		

Conclusion

RECENT POSTS

Building .NET projects using the Microsoft.Build.Traversal SDK

Tracking Cloudflare Worker requests using Application Insight

Using raw html with isolated CS in Blazor

Be careful when mixing ValueTa and Task.Run

Rewriting git history using C# (g filter–branch alternative)

LINKS

Dev Tool List

Dev News

Async/await resources

.NET Multithreading resources

Online tools

Using HTTP caching is important for performance reasons (client and server sides). With ASP.NET Core, you can take advantage of the provided `TagHelper` s to generate a versioned URL, and change the default configuration of the `StaticFilesMiddleware` to add the `Cache-control` header for these URLs.

Do you have a question or a suggestion about this post? Contact me!

Follow me:

Enjoy this blog? Buy me a coffee