

Change the headers of static files in Asp.net Core

Asked 6 years ago Active 2 years, 6 months ago Viewed 7k times



13



3



I am using package `Microsoft.AspNet.StaticFiles` and configuring it in `Startup.cs` as `app.UseStaticFiles()` . How can I change the headers of the delivered files ? I want to set cache expiry etc for images, css and js.

[asp.net-core](#) [asp.net-core-mvc](#)

Share Improve this question Follow

edited May 16 '16 at 5:03

asked Mar 25 '15 at 12:53



[eadam](#)

15.4k

15

41

69

1 [Caching static resources forever with ASP.NET Core](#) – [meziantou](#) Feb 28 '17 at 10:04

5 Answers

Active

Oldest

Votes



20



You can use `StaticFileOptions`, which contains an event handler that is called on each request of a static file.

Your `Startup.cs` should look something like this:

```
// Add static files to the request pipeline.
app.UseStaticFiles(new StaticFileOptions()
{
    OnPrepareResponse = (context) =>
    {
        // Disable caching of all static files.
        context.Context.Response.Headers["Cache-Control"] = "no-cache, no-store";
        context.Context.Response.Headers["Pragma"] = "no-cache";
        context.Context.Response.Headers["Expires"] = "-1";
    }
});
```

You can, of course, modify the above code to check the content type and only modify headers for JS or CSS or whatever you want.

Share Improve this answer Follow

edited Jan 11 '16 at 15:52

answered Jan 11 '16 at 15:45



[Josh Mouch](#)

3,209

1

32

31

Join **Stack Overflow** to learn, share knowledge, and build your career.

Sign up



[HeaderNames.CacheControl] = "public, max-age=86400"; (86400 here is 24 hour = 24*60*60 sec). And to force bypass the cache when we update the files, We can also use `asp-append-version="true"` tag helper beside the `<link>`, `<script>` ... tags to add some auto generated query string based on hash of the files which will be auto-updated. – [S.Serpooshan](#) Oct 21 '18 at 8:05

Based on Josh Mouch's answer above, added code to determine if it's a pdf file

6 Startup.cs:

```
app.UseStaticFiles(new StaticFileOptions
{
    OnPrepareResponse = ctx =>
    {
        if(ctx.File.Name.ToLower().EndsWith(".pdf"))
        {
            ctx.Context.Response.Headers.Append("Cache-Control", "public,max-age=86400");
        }
        else
        {
            ctx.Context.Response.Headers.Append("Cache-Control", "public,max-age=31104000");
        }
    }
});
```

Share Improve this answer Follow

answered Jul 13 '17 at 17:52



[Matt Lengenfelder](#)

659 5 15

You have to write a middleware to do this, I've a sample that remove headers on my github

<https://github.com/aguacongas/chatle>

2 Look at the ChatLe.HttpUtility project, it's a bit tricky. you can take a look of this question as well:

[How to do remove some httpresponse headers on each response like Server and ETag?](#)

However this will not work under IIS, because IIS manage statics files itself. It will work only on stand-alone application like kestrel or firefly

Share Improve this answer Follow

edited May 23 '17 at 12:09



Community ♦

1 1

answered Mar 25 '15 at 14:17



[agua from mars](#)

12.6k 4 45 55

Do you mean all three of those won't work under IIS? Or one of the answers, specifically? – [Josh Mouch](#) Jan 11 '16 at 15:31

2 **web.config** file instead of hard-coding the whole stuff, you could consider the following approach.

Add the following key/value section in the **appsettings.json** file:

```
"StaticFiles": {
  "Headers": {
    "Cache-Control": "no-cache, no-store",
    "Pragma": "no-cache",
    "Expires": "-1"
  }
}
```

Then add the following in the **Startup.cs** file's `Configure` method accordingly:

```
app.UseStaticFiles(new StaticFileOptions()
{
    OnPrepareResponse = (context) =>
    {
        // Disable caching for all static files.
        context.Context.Response.Headers["Cache-Control"] =
Configuration["StaticFiles:Headers:Cache-Control"];
        context.Context.Response.Headers["Pragma"] =
Configuration["StaticFiles:Headers:Pragma"];
        context.Context.Response.Headers["Expires"] =
Configuration["StaticFiles:Headers:Expires"];
    }
});
```

This will allow the developer to define different cache settings using different/multiple/cascading settings files (`appsettings.json` , `appsettings.production.json` and so on) - which is something that could be done with the old `web.config` configuration pattern - with the ASP.NET Core's new one.

For additional info regarding the topic I also suggest to read [this post](#) on my blog and/or these great articles from the official ASP.NET Core docs:

- [Configuration](#)
- [Working with Static Files](#)

Share Improve this answer Follow

edited Oct 19 '18 at 15:07

answered Aug 5 '16 at 17:43



Darkseal

8,258 6 67 95

Under IIS, you can add a `web.config` file to your `wwwroot` folder with header configurations. An example that will control cache headers for all files:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Sign up



```
<httpProtocol>
  <customHeaders>
    <add name="Cache-Control" value="no-cache" />
  </customHeaders>
</httpProtocol>

</system.webServer>
</configuration>
```

Share Improve this answer Follow

answered Apr 7 '15 at 20:28



Malgaur

1,762 13 17

- 1 That won't work if the websites host .NET Core modules (like in case of OP), as .NET Core doesn't honour web.config . The way .NET Core works is essentially running their own web server called Kestrel, which is hiding behind IIS + HTTP Module (aka ANCM, aka ASP.NET Core Module). docs.microsoft.com/en-us/aspnet/core/fundamentals/servers/... – Alex Klaus Feb 1 '17 at 6:46