Output cache middleware in .NET 7

Ajai K M

Lead Engineer - Suyati Technologies Pvt Ltd

https://www.linkedin.com/in/ajai-km/

Agenda

- Introduce the .Net7 output cache middleware
- Enable output cache
- Caching properties
- Clear/Flush/Expire cached values before expiry
- Create custom cache policy
- References

What is output caching

- Output caching helps you store results from your web app and web API and serve them from a cache rather than computing them every time,..
- which improves performance and frees up resources for other activities.
- In output cache rather than checking HTTP cache headers to decide what should and should not be cached,..
- the decision is instead made directly by the application, likely with a new request Feature, attributes, HttpResponse extension methods, etc.
- This cache also does not honor any request cache headers from the client.
- The client isn’t supposed to know it’s receiving a cached response.

Do’s and Don'ts

- Avoid caching contents that are unique per user.
- Avoid caching contents that are accessed rarely.
- Use caching for contents that are accessed frequently.
- Enable caching for a page when the page does not require authorization.

Enable output caching

For enable Output caching, you need to add its middleware to service collection in **program.cs**

```
1   builder.Services.AddOutputCache();
2   var app = builder.Build();
3   app.UseOutputCache();
```

now we must create our cache policy to middleware by

AddBasePolicy() and **AddPolicy()** methods:

AddBasePolicy & AddPolicy

- **AddBasePolicy:** This method set base behaviors for all policies. Also, we can override this behavior in each policy.
- **AddPolicy:** This method set a policy. All policies inherits behavior from AddBasePolicy method . We can override any behavior inherited from base policy by set it again in AddPolicy, for example we can set expire time to 5 min. For each policy we must set name and use this name in place to have policy for intended action.

```
builder.Services.AddOutputCache(options =>
{
    options.AddBasePolicy(policy => policy
        .Expire(TimeSpan.FromMinutes(10)));

    options.AddPolicy("PeoplePolicy", policy => policy
        .Expire(TimeSpan.FromMinutes(5)));
}
```

Caching properties

We can cache by route values, query string, request headers or set not to store response or even change cache duration. For each one we can set like what is set for PolicyName.

- VaryByRouteValueNames : read and set from route.
- VaryByHeaderNames : read and set from header.
- VaryByQueryKeys : read and set from query
- NoStore : set not to store response (is boolean)
- Duration : change or set cache duration (in seconds)

Clear/Flush/Expire cached values

For clear/flush/expire cached values before expire time we can use EvictByTagAsync() method of IOutputCacheStore. this method take "Tag" and cancellation token as parameter and clear cache values.

so, for clearing a cached value (values) we have to set Tag to policy. we can set it as follow:

```
builder.Services.AddOutputCache(options =>
{
  options.AddPolicy("PeoplePolicy", policy => policy
        .Expire(TimeSpan.FromMinutes(10))
        .Tag("PeoplePolicy_Tag"));
}
```

Custom cache policy

Imagine we have person and skill entities, and each person can have multiple skill, now we want cache each person and his/her skills by Person Id. In this situation Tag that we use before doesn't work properly because it clear all cached values.

At time of writing this skill there is no any method or function introduced by **OutputCache** that can do it straight forward. So, we have to create custom policy. to achieve this aim, we must implement **IOutputCachePolicy**

register it as follow:

```
1  builder.Services.AddOutputCache(options =>
2  {
3      options.AddPolicy("ByIdCachePolicy", policy => policy
4      .AddPolicy<ByIdCachePolicy>()
5      .Expire(TimeSpan.FromMinutes(5)));
6  });
```

Reference

- https://learn.microsoft.com/en-us/aspnet/core/performance/caching/output?view=aspnetcore-7.0
- https://timdeschryver.dev/blog/exploring-the-new-output-caching-middleware
- https://alirezafarokhi.medium.com/net-7-outputcache-and-custom-outputcache-c01e6c9d8c