Supporting HTTP Cache for ASP.NET Core APIs

Kevin Dockx
Architect

@KevinDockx https://www.kevindockx.com

Coming Up

Working with ETags

- Global configuration
- Resource-level configuration

Cache stores and CDNs

Cache invalidation

Supporting ETags

ETags are preferred over dates as they are strong validators

Supporting ETags

Marvin.Cache.Headers

- https://github.com/KevinDockx/HttpCacheHeaders
- ASP.NET Core middleware that adds HTTP cache headers to responses, like `Cache-Control`, `Expires`, `ETag` and `Last-Modified`
- Implements cache expiration & validation models

Demo

Adding support for generating ETags

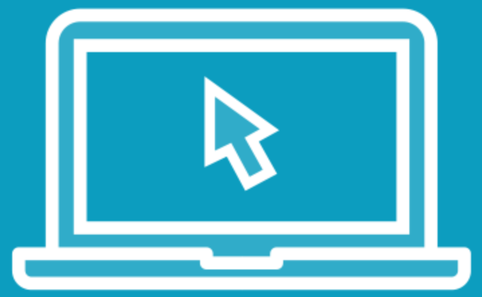
Demo

Global cache header configuration

Demo

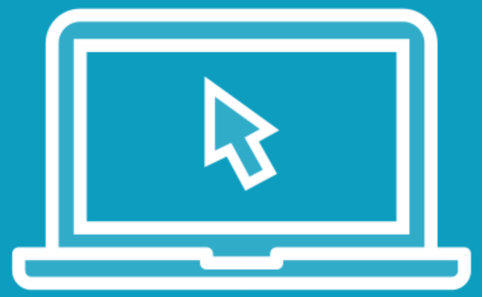
Resource-level cache header configuration

Demo



Dealing with varying response representations

Demo



ETags and the validation model

Cache Stores and Content Delivery Networks

Most cache stores are full-blown cache servers, not pieces of middleware

Private Caches

Live on the client

- UWP apps, WPF apps, ...

CacheCow.Client

- https://github.com/aliostad/CacheCow

Shared Caches

Gateway or proxy caches

Full-blown cache servers

- Varnish
  - https://varnish-cache.org/
- Apache Traffic Server
  - http://trafficserver.apache.org/
- Squid
  - http://www.squid-cache.org/

Content Delivery Network

A content delivery network (CDN) is a system of distributed servers (network) that deliver pages and other web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.

Content Delivery Network

Most of the internet runs on various CDNs

CDNs extensively use caching: HTTP cache

- No need to set up a cache server ourselves

Content Delivery Network

Popular CDN examples

- Azure CDN
  - https://azure.microsoft.com/en-in/services/cdn/
- Cloudflare
  - https://www.cloudflare.com
- Akamai
  - https://www.akamai.com

Cache Stores and Content Delivery Networks

Ensure your API can return `Cache-Control` headers and supports expiration/validation models

Combine that with a cache server or CDN

Cache Invalidation

Wiping a response from the cache because you know it isn't the correct version anymore

Cache Invalidation

A lot of this is automated

- Responses go stale
- ETags get updated

Cache Invalidation

But that's not always sufficient

- Resource manipulation can have an effect on other resources
  - Updating a course affects the courses resource
  - Same goes for deleting or creating a course resource

Cache Invalidation

CDNs allow for cache invalidation via easy to configure rules and easy to use SDKs

Most cache servers also support an invalidation mechanism

Summary

Use strong validators like ETags when supporting the validation model

Shared caches are often full-blown cache servers

- Varnish, Apache Traffic Server, Squid, ...

CDNs provide cache servers out of the box

Up Next:
Supporting Concurrency