

# Cacheování v .NETu i HTTP

Michal Altair Valášek



Altairis, s. r. o.

[www.altair.blog](http://www.altair.blog)

[michal.valasek@altairis.cz](mailto:michal.valasek@altairis.cz)

Tech·Ed

 GOPAS



# Možnosti cacheování v ASP.NET Core

## Object caching

- Není webový, umožňuje cacheovat libovolné objekty
- In-memory nebo distributed cache

## Razor tag helpers

- Postavené nad object cachingem
- Umožňují cacheovat části view v MVC a Razor Pages

## Output caching<sup>7.0</sup>

- Cacheování na straně serveru
- Je vynutitelné nezávisle na vůli klienta

## Response caching

- Řeší cacheování na straně klienta
- Pouze nastavuje hlavičky HTTP odpovědi

# Object cache

# Object cache

- Umožňuje cacheovat objekty v rámci infrastruktury
  - Např. výsledky db dotazu nebo API volání
- Nesouvisí s webem ani ASP.NET Core
- **IMemoryCache**
  - In-memory implementace
  - Drží se v paměti procesu
  - Nemůže být sdílená mezi servery
- **IDistributedCache**
  - Externí cache
  - Přežije restart
  - Může být sdílená

# Použití IMemoryCache

- Nainjectujte si ji tam, kde je třeba
- Použijte nejlépe metody `GetOrCreate` a `GetOrCreateAsync`
  - Pokud položka existuje, vrátí ji z cache
  - Pokud neexistuje, vytvoří a uloží ji tam
- Tyto metody jsou thread safe
- Negarantují ale, že se vytvoření zavolá pouze jednou!
  - Pokud je to problém, použijte wrapper `LazyCache`
  - <https://github.com/alastairtree/LazyCache>

# DEMO

IMemoryCache

# Expiration (Change) Tokens

- `IChangeToken`
  - Vestavěný mechanismus v .NET pro sledování změn obecně
- Lze jej použít pro vyhození položky v cache v závislosti na změně
- Přidejte pomocí metody `AddExpirationToken`
- Typicky se používá závislosti na souboru
  - Tam lze získat pomocí `IFileProvider.Watch(fileName)`

# DEMO

File dependency



# Použití IDistributedCache

- Umožňuje použití externí cacheovací služby
- Může být sdílená mezi více servery
- Implementace v .NET
  - Distributed Memory cache (výchozí, není ve skutečnosti distribuovaná)
  - Distributed SQL Server cache
  - Distributed Redis cache
  - Distributed NCache cache
  - + další implementace třetích stran

# DEMO

IDistributedCache

# Razor Cache Tag Helpers

# Razor Cache Tag Helpers

- Umožňují cacheovat část HTML výstupu
- Použitelné v Razoru (MVC, Razor Pages)
- `<cache>`
  - In-memory cache
- `<distributed-cache>`
  - Distributed cache, s in-memory cache jako výchozí implementací
  - Doporučuji používat, protože otevírá cestu k budoucímu rozšíření
- Řeší jenom view infrastrukturu! Ne model!
  - V modelu typicky použijte `Lazy<T>`

# Atributy elementů `<cache>` a `<distributed-cache>`

- `name` klíč (*pouze pro distributed-cache*)
- `enabled` `true` nebo `false`
- `expires-after` absolutní exp. `TimeSpan`
- `expires-on` absolutní exp. `DateTime`
- `expires-sliding` relativní exp. `TimeSpan`
- `vary-by(-*)` viz dále

# Varianty cache

- `vary-by` libovolný string (z modelu)
- `vary-by-cookie` seznam názvů cookies oddělený čárkami
- `vary-by-culture` `true` nebo `false`
- `vary-by-header` názvy HTTP hlaviček
- `vary-by-query` názvy query parametrů
- `vary-by-route` názvy route parametrů
- `vary-by-user` `true` nebo `false`

# DEMO

Razor Tag Helpers pro cache

# Output Cache



# Output cache

- Novinka v ASP.NET 7.0
- Cacheování HTTP odpovědi
  - Middleware použitelný kdekoliv
  - Razor Pages, MVC, WebAPI, Minimal APIs, ...
- Fyzicky se drží v in-memory cache

# Použití output cache

- Atribut (MVC, Razor Pages, Web API) nebo metoda (Minimal APIs) `OutputCache`
- Umožňuje nastavit parametry podobně jako cache tag helpers (vary-by, expirace...)
  - Parametry lze nastavit přímo ad hoc
  - Lepší je ale definovat cache policy a pak se na ni odkázat jménem
- Umožňuje tagging
  - Položky lze otagovat
  - Pak lze odstranit položky s určitým tagem

# DEMO

Output cache

# Response Caching

# Response caching

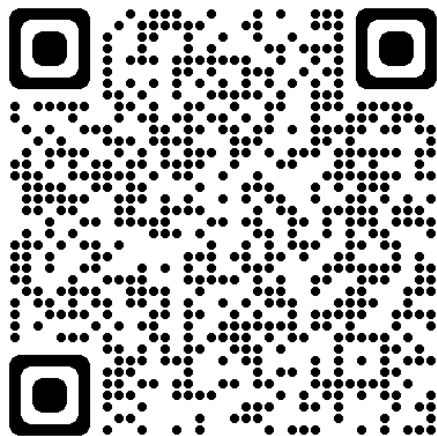
- Ve skutečnosti nic necacheuje
  - Pouze nastavuje HTTP hlavičky (zejména Cache-Control)
  - Vlastní cacheování provádí browser nebo mezilehlá proxy
  - Pokud je nemáte pod kontrolou, nelze si cache vynutit
- Uživatelsky přítulné nastavení HTTP hlaviček
  - Funguje v MVC, Web API a Razor Pages
  - Kdekoliv jinde lze s hlavičkami pracovat přímo
- Lze nastavovat přímo nebo (lépe) pomocí pojmenovaných profilů
  - Připomíná to policies z Output Cache
- Typicky kombinujeme s dalšími metodami cacheování

# Podrobněji o cacheování



Seriál ASP.NET Cache na kanálu Z-TECH:

<https://www.youtube.com/playlist?list=PLFZurxJN0pMbTTXyhI7SQIEEr04YIyIPG>



✉ [michal.valasek@altairis.cz](mailto:michal.valasek@altairis.cz)

f [facebook.com/rider.cz](https://facebook.com/rider.cz)

g [github.com/ridercz](https://github.com/ridercz)

t [twitter.com/ridercz](https://twitter.com/ridercz)

in [linkedin.com/in/ridercz](https://linkedin.com/in/ridercz)

📄 [www.altairis.cz](http://www.altairis.cz)

[www.rider.cz](http://www.rider.cz)

[www.altair.blog](http://www.altair.blog)

**Tech·Ed**

 **GOPAS**

