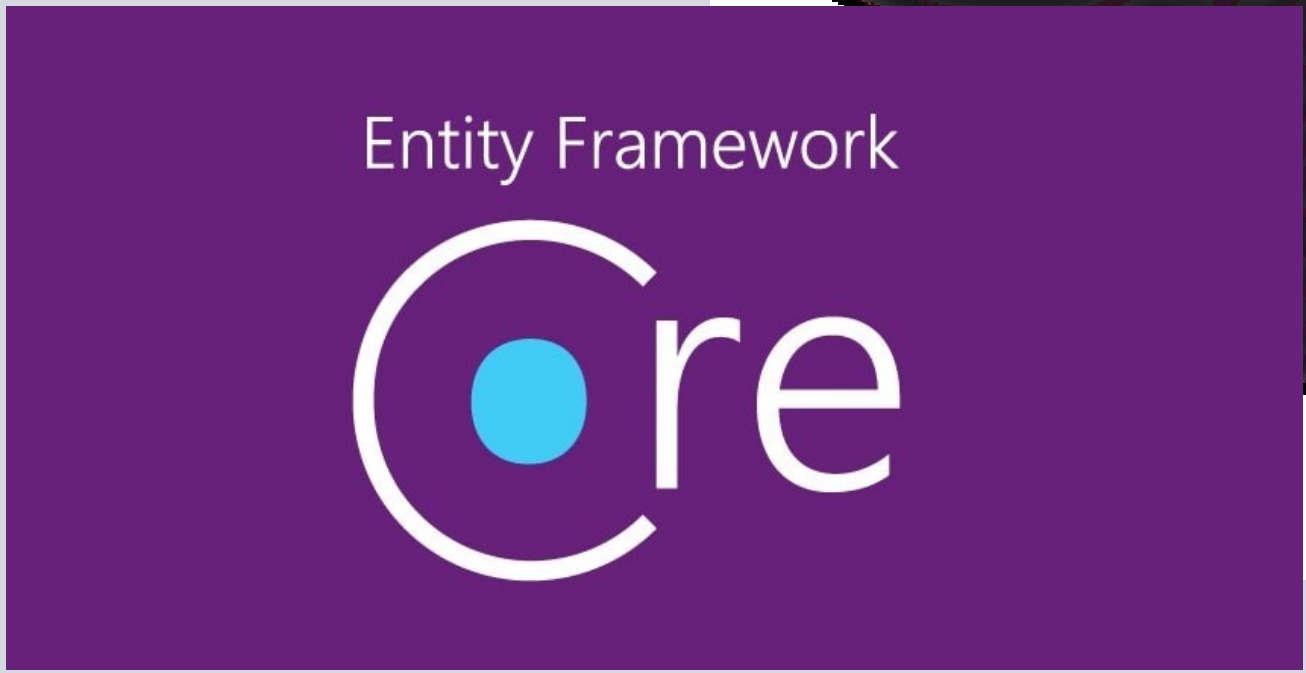
Ukládání a výměna dat na webu

Ondřej Pavlica

Trvalá data

- Data, která jsou důležitá v dlouhodobějším horizontu / nesmí se ztratit.
- Databáze, filesystém, cloud (např. object storage)
- Můžeme využít EF Core (relační DB – SQL Server, PostgreSQL, …), MongoDB, Amazon S3, …
- Nejčastější use case pro využití repository patternu



Výměna dat v rámci ASP.NET Core aplikace

- Cache – služba `IMemoryCache`, pozor na kapacitu a timeouty!
- Mezi controllerem a view:
  - `Model` – dobré pro jednu stránku, problém při obecných věcech (Layout)
  - `ViewData` – sdílený slovník předaný z controlleru do view (včetně partial views), opačným směrem však ne
  - `HttpContext.Items` – sdílené mezi všemi kroky v rámci jednoho requestu (tzn. Middleware, controller, view, atributy, atd.)
    - K `HttpContext` jde přistoupit, i když není dostupný jako property, přes službu `IHttpContextAccessor`
    - `IHttpContextAccessor` je potřeba zaregistrovat při startu aplikace přes `builder.Services.AddHttpContextAccessor()`

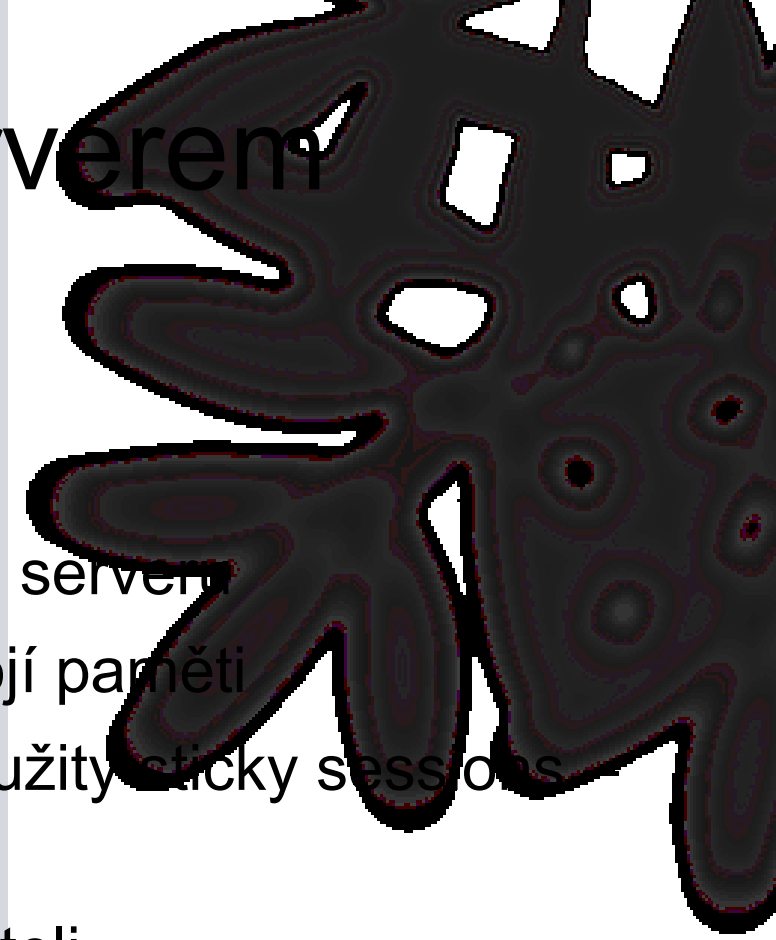
Výměna dat mezi klientem a serverem

- Jako součást requestu – vhodné pro jednorázová data:
  - URL parametry (klient -> server) – přístup přes model binding
  - Formulářová data (klient -> server) - přístup přes model binding
  - HTTP headery (obousměrné) – přístup přes model binding či `Request.Headers[„headerName“]`, zápis přes `Response.Headers[„headerName“]`

Výměna dat mezi klientem a serverem (pokr.)

- Dlouhodobější data – cookies - obousměrná
  - Cookies lze na klientovi zobrazit a změnit – potřeba chránit citlivé údaje a data z cookies ověřovat na straně serveru
  - Přístup přes `Request.Cookies[„cookieName“]`, zápis přes `Response.Cookies[„cookieName“]`
  - Hodnota cookie je string, správně by měl být kódovaný
    - URL kódování pro plaintext
    - Base64 pro ostatní data
  - Většina prohlížečů limituje velikost cookies na 4 kB

Výměna dat mezi klientem a serverem (pokr.)



- Krátkodobější data – session state
  - Klient zná pouze své session ID, které posílá v cookie serveru
  - Ten si pomocí tohoto ID napáruje správná data ze svojí paměti
  - Není unikátní pro jednoho uživatele (pokud nejsou použity sticky sessions, omezuje škálovatelnost) ani okno
  - Nejčastěji používané jako cache údajů o daném uživateli
  - Nastavuje se přes `builder.Services.AddDistributedMemoryCache()` plus `builder.Services.AddSession()`
  - Přístup přes `HttpContext.Session.GetString() / SetInt32()`, apod.

Výměna dat mezi serverem a klientem

- REST API – data jsou obsahem odpovědi serveru
- MVC
  - Postranní kanál – header, cookie – nedoporučeno pro základní use case
  - Součást vyrenderovaného HTML
    - Data čistě k zobrazení klientovi:
      - Podmíněně vyrenderované kus HTML (např. hláška o úspěchu akce)
    - Data k předání JS kódu:
      - Inicializace dat v rámci JS v `<script>` bloku vygenerovaným serverem
      - Použití data atributů, např `<div id=‘save-form‘ data-localized-succes-message=‘Úspěšně uloženo‘>`
        - Přístup v JS přes `document.getElementById(‘save-form‘).dataset.localizedSuccessMessage`

Komplexnější obousměrná komunikace

- Mimo rozsah cvičení
- Většinou pomocí WebSockets a nebo pollingu přes HTTP dotazy
- Knihovny třetích stran, např. SignalR, gRPC