

# Zarządzanie sesją w ASP.NET Core

Programowanie aplikacji WWW w technologii .NET, 2021/2022

Przygotowała: I. Kartowicz-Stolarska

## CEL

Rozszerzenie funkcjonalności aplikacji FizzBuzzWeb o:

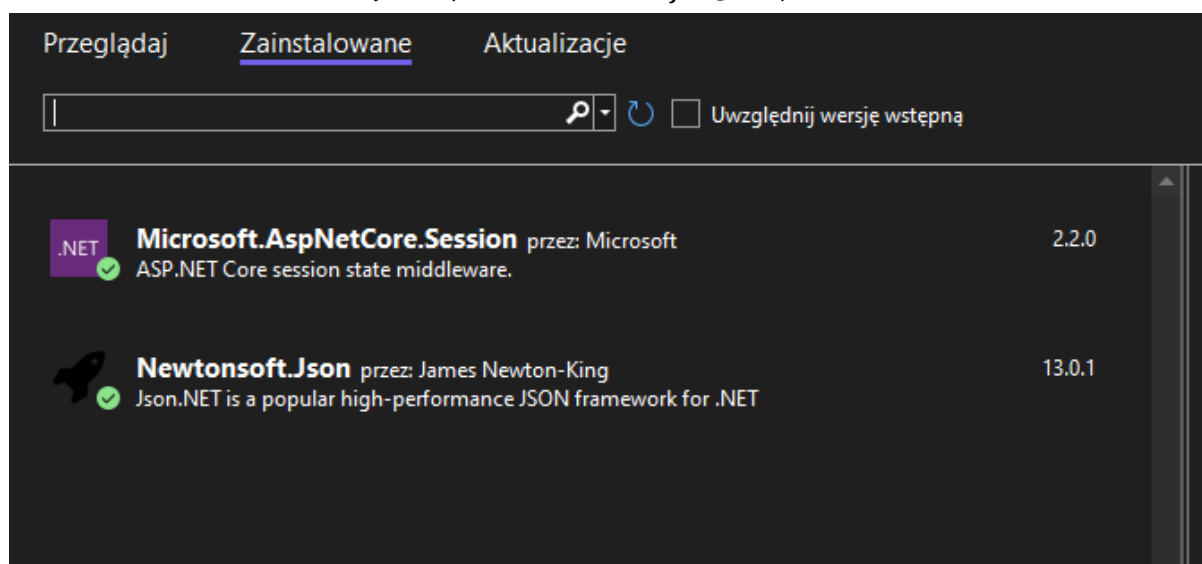
- zapis danych w sesji; dane przesyłane są przez formularz prezentowany na stronie głównej (szablon Index.cshtml),
- odczyt danych pobranych z sesji na nowej stronie Razor.

## WYMAGANIA

1. Zainstalowany pakiet NuGet Microsoft.AspNetCore.Session
2. Włączone SessionState w Program.cs / w Startup.cs (dla .Net Core < 6)
3. Przestrzeń nazw: Microsoft.AspNetCore.Http

## TUTORIAL

1. Otwórz aplikację FizzBuzz z poprzednich zajęć.
2. Zainstaluj pakiety Nuget (Projekt->Zarządzaj pakietami NuGet):
  - a. Microsoft.AspNetCore.Session (dla .Net 6 wersja 2.2.0)
  - b. Newtonsoft.Json (dla .Net 6 wersja 13.0.1)



Skorzystaj z widoku Projekt->Zarządzanie pakietami NuGet (Microsoft Visual Studio) lub terminala (Microsoft Visual Studio/Visual Studio Code):

```
dotnet add package Microsoft.AspNetCore.Session
```

```
dotnet add package Newtonsoft.Json
```

3. Włącz obsługę sesji. Zwróć uwagę na kolejność wstrzykiwania serwisów i włączania modułów.

- a. dla .Net 6:

- i. w *Program.cs* dodaj:

```
var builder = WebApplication.CreateBuilder(args);  
// Add services to the container.  
builder.Services.AddRazorPages();  
builder.Services.AddMemoryCache();  
builder.Services.AddSession();  
...  
app.UseAuthorization();  
app.UseSession();  
app.MapRazorPages();  
...
```

- b. dla wersji .Net Core 3.1 lub .Net 5:

- i. w *Startup.cs*:

```
public void ConfigureServices(IServiceCollection services){  
    ...  
    services.AddMemoryCache();  
    services.AddSession();  
}  
public void Configure(IApplicationBuilder app,  
    IWebHostEnvironment env){  
    ...  
    app.useSession();  
    ...  
}
```

4. Zamiast `AddMemoryCache` możesz użyć `AddDistributedMemoryCache`. Czym różnią się te dwa typy pamięci. W jakich projektach powinieneś/powinnaś używać pamięci rozproszonej?

- .Net 6:

- `builder.Services.AddDistributedMemoryCache();`

- .Net Core 3.1 i .Net 5:

- `services.AddDistributedMemoryCache();`

5. W `builder.Services.AddSession()/service.AddSession()` możesz ustawić dodatkowe parametry np.

```
builder.Services.AddSession(options =>  
{  
    options.IdleTimeout = TimeSpan.FromSeconds(10);  
    options.Cookie.HttpOnly = true;
```

```
options.Cookie.IsEssential = true;
});
```

6. Dodaj nową stronę Razor do projektu (kliknij w Pages, a następnie z menu kontekstowe wybierz Dodaj->Strona Razor) np. o nazwie *SavedInSession.cshtml*.

Możesz też użyć polecenia:

```
dotnet new page -n SavedInSession -na FizzBuzzWeb.Pages -o Pages
```

W obu przypadkach w katalogu Pages zostanie dodany nowy model i szablon strony o nazwie *SavedInSession*.

7. Dodaj nowy element nawigacji do pliku *Pages/Shared/\_Layout.cshtml*

```
...
<li class="nav-item">
    <a class="nav-link text-dark" asp-area=""
    asp-page="/SavedInSession">Saved in session</a>
</li>
...
```

## Zapis do sesji.

8. W pliku *Pages/Index.cshtml/Index.cshtml.cs* dodaj przestrzeń nazw:
 

```
using Microsoft.AspNetCore.Http;
using Newtonsoft.Json;
```
9. W pliku *Pages/Index.cshtml/Index.cshtml.cs* zmodyfikuj metodę *onPost()* w taki sposób, aby zapisywała obiekt *FizzBuzz* do sesji pod nazwą *Data* tylko wtedy, gdy dane podane przez użytkownika strony są prawidłowe.

```
public IActionResult OnPost() {
    if (ModelState.IsValid) {
        HttpContext.Session.SetString("Data",
        JsonConvert.SerializeObject(FizzBuzz));
        return RedirectToPage("./SavedInSession");
    }
    return Page();
}
```

## Odczyt z sesji.

10. W pliku *SavedInSession.cshtml.cs* dodaj nowy atrybut *FizzBuzz* oraz uzupełnij metodę *onGet()* o odczyt danych z sesji:

```
using FizzBuzzWeb.Models;
```

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.AspNetCore.Http;
using Newtonsoft.Json;

namespace FizzBuzzWeb.Pages
{
    public class SavedInSessionModel : PageModel
    {
        public FizzBuzz FizzBuzz { get; set; }

        public void OnGet()
        {
            var Data = HttpContext.Session.GetString("Data");

            if (Data != null)
                FizzBuzz =
                    JsonConvert.DeserializeObject<FizzBuzz>(Data);
        }
    }
}

```

#### 11. Zmodyfikuj szablon strony *Pages/SavedInSession.cshtml* o wyświetlanie danych pobranych z sesji:

```

@page
@model SavedInSessionModel
@{
    ViewData["Title"] = "Data in session";
}
<p>Dane w zapisane w sesji</p>
@if (Model.FizzBuzz != null)
{
    <p>@Html.DisplayNameFor(model => Model.FizzBuzz.Number):
    @Model.FizzBuzz.Number </p>
}

```

#### 12. Bazując na dokumentacji

<https://docs.microsoft.com/pl-pl/aspnet/core/fundamentals/app-state?view=aspnetcore-6.0>

- sprawdź jak działa `HttpContext.Session.SetInt32()` i `HttpContext.Session.GetInt32()`
- spróbuj zserializować listę obiektów np. listę ostatnio podanych liczb i rezultatów sprawdzania FizzBuzz.
- spróbuj odczytać listę obiektów

## REFERENCJE

1. <https://docs.microsoft.com/pl-pl/dotnet/core/tools/dotnet-add-package>
2. <https://docs.microsoft.com/pl-pl/aspnet/core/fundamentals/app-state?view=aspnetcore-6.0>
3. <https://docs.microsoft.com/pl-pl/aspnet/core/performance/caching/memory?view=aspnetcore-6.0>
4. <https://docs.microsoft.com/pl-pl/aspnet/core/performance/caching/distributed?view=aspnetcore-6.0>

## ZADANIE DOMOWE

Wykonaj aplikację webową, która będzie sprawdzała, czy podany przez użytkownika rok urodzenia jest/był rokiem przestępnym.

Założenia:

- Aplikacja składa się z dwóch stron: Index i Zapisane.
- Na głównej stronie użytkownik odwiedzający aplikację widzi formularz mający dwa pola:
  - rok  
Zakres wartości od 1899 do 2022. Jeśli użytkownik poda rok spoza zakresu lub poda inny typ danych np. znaki, wyświetlany jest komunikat o błędzie.
  - imię użytkownika  
ograniczenie do 100 liter.

Oba pola są wymagane. Błędy oznaczone są na czerwono, pogrubioną czcionką.

- Na stronie głównej użytkownik otrzymuje informację o wyniku sprawdzania, w formacie jak niżej:
  - Franek urodził się w 2020 roku. To był rok przestępny.
  - Zośka urodziła się w 2001 roku. To nie był rok przestępny.
- Wyniki wszystkich sprawdzeń są zapisywane w sesji.
- Na stronie "Zapisane" wyświetlana jest lista wszystkich sprawdzeń w postaci:
  - Franek, 2020 - rok przestępny
- Projekt powinien zostać opublikowany w repozytorium Git.
- Link do repozytorium proszę umieścić na platformie CEZ.