

# Filtrowanie

*Programowanie aplikacji WWW w technologii .NET, 2021/2022*

*Przygotowała: I. Kartowicz-Stolarska*

## CEL

1. Obsługa filtrów w .Net 6 Razor Page na przykładzie filtra atrybutów.

## BAZA WIEDZY

Zapoznaj się z artykułami i video:

- <https://docs.microsoft.com/en-us/aspnet/core/razor-pages/filter?view=aspnetcore-6.0>
- <https://www.mikesdotnetting.com/article/323/accessing-the-model-from-filters-in-razor-pages>
- <https://www.youtube.com/watch?v=ayNJieQQFKU>

## TUTORIAL

1. Utwórz nowy projekt.
2. Utwórz folder o nazwie Utlis oraz nową klasę AddHeaderAttribute.
3. W klasie AddHeaderAttribute dodaj dziedziczenie po klasie ResultFilterAttribute.
4. Uzupełnij kod klasy AddHeaderAttribute:

```
private readonly string _name;
private readonly string _value;

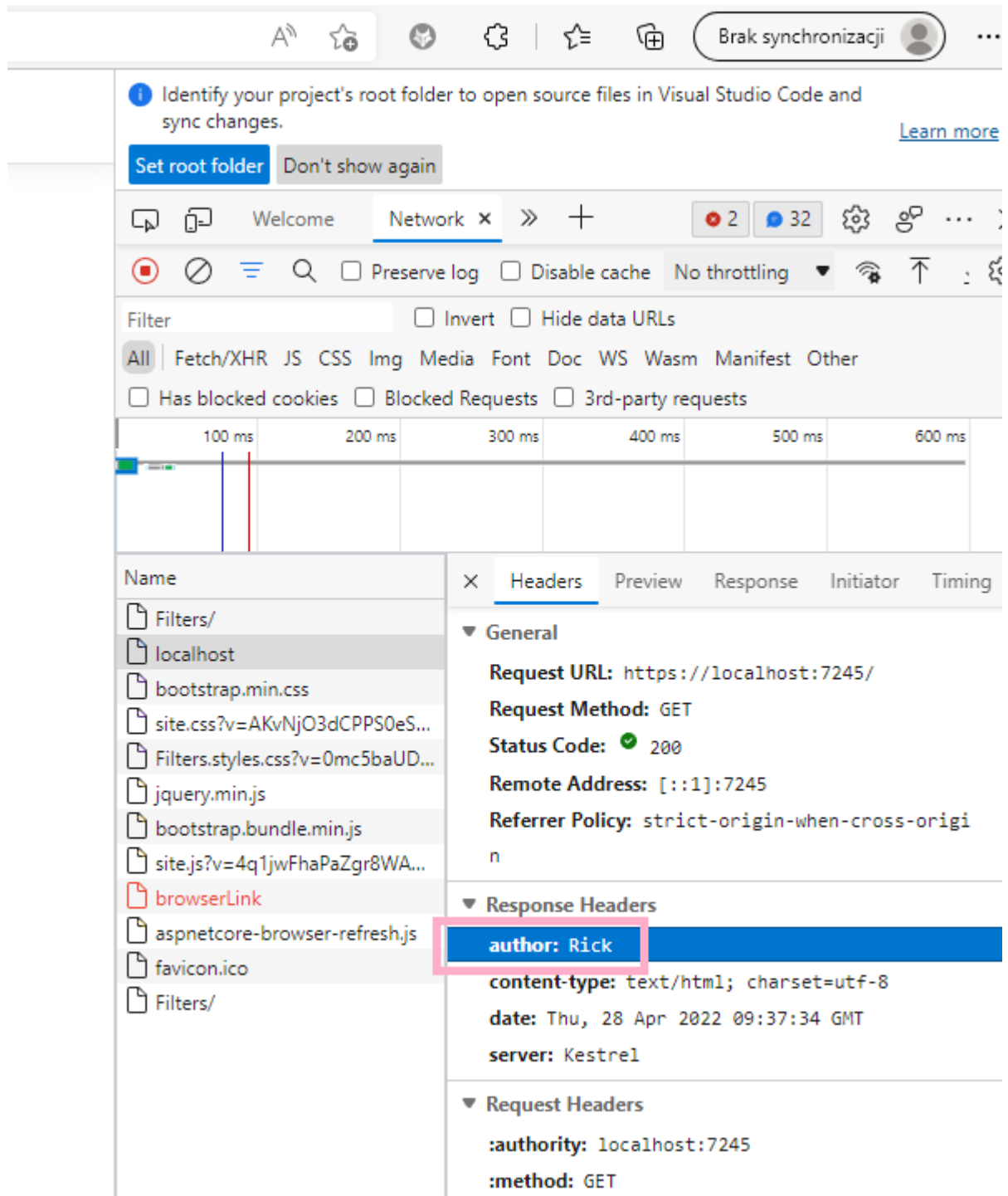
public CustomFilllterAttr(string name, string value) {
    _name = name;
    _value = value;
}

public override void OnResultExecuting(ResultExecutingContext context) {
    context.HttpContext.Response.Headers.Add(_name, new
string[] { _value });
}
```

5. W IndexModel wywołaj filtr:

```
[AddHeaderAttribute("Author", "Rick")]
public class IndexModel : PageModel
{
    ...
}
```

- Uruchom projekt i sprawdź, czy po wywołaniu IndexModel w przeglądarce pojawił się nowy nagłówek:



- Realizuj tutorial z artykułu <https://www.learnrazorpages.com/razor-pages/filters> realizujący filtry globalne dla stron Razor oraz filtry atrybutów. Wstaw punkty przerwania w

konkretnych implementacjach filtrów i sprawdź kolejność wywołania filtrów.

Uwagi do implementacji:

- Wywołanie filtrów w Program.cs

```
builder.Services.AddMvc(options =>
{
    options.Filters.Add(new CustomPageFilter());
});
builder.Services.AddTransient<IGeoService, GeoService>();
```

- Metodę GetCountry w serwisie GeoService zmień:

```
public async Task<string> GetCountry(string ipAddress)
{
    using (var client = new HttpClient())
    {
        var info = await
client.GetStringAsync($"http://edns.ip-api.com/");
        return info;
    }
}
```

8. Do klasy PrivacyModel dodaj dwie metody:

```
public override void
OnPageHandlerSelected(PageHandlerSelectedContext context)
{
    int i = 0;
}

public override void
OnPageHandlerExecuting(PageHandlerExecutingContext context)
{
    int i = 1;
}
```

Za pomocą punktów przerwania prześledź kolejność wykonania tych metod przez framework .NET.