

Mariusz Wieczorek

mariusz.wieczorek@kabat.pl

Repozytorium, UnitOfWork, Serwisy

Jak w praktyce użyć tych wzorców

asp.net Core

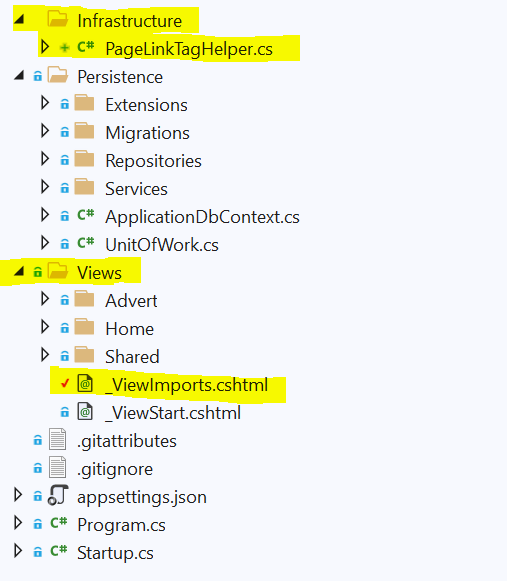
Klasa Atrybutu Pomocniczego

**Dodanie klasy atrybutu pomocniczego**

Tworzymy katalog **Infrastructure** - w nim przechowujemy klasy komponentów aplikacji nie powiązanych z domeną

W tym katalogu tworzymy plik: PageLinkTagHelper.cs

Atrybuty pomocnicze znaczników w odróżnieniu od widoków i kontrolerów muszą być rejestrowanie robimy to w pliku: Views\\_ViewImports.cshtml



|  |
| --- |
| @using MyAdvert  @using MyAdvert.Models  @using MyAdvert.Core.ViewModels  @addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers  @addTagHelper MyAdvert.Infrastructure.\*, MyAdvert |

**Dodawanie Stronicowania ( paginacji )**

Aby zapewnić możliwość prawidłowego działania atrybutów pomocniczych znaczników musimy przekazać informacje o liczbie dostępnych stron, bieżącej stronie oraz całkowitej liczbie pozycji w repozytorium.

Najprostszym sposobem zrealizowania tego zadania jest utworzenie modelu widoku, który będzie używany do przekazywania danych między kontrolerem i widokiem.  
Tworzymy katalog Models/ViewModels i tworzymy w nim plik klasy PagingInfo.cs

|  |
| --- |
| namespace MyAdvert.Core.ViewModels  {  public class PagingInfo  {  public int TotalItems { get; set; }  public int ItemsPerPage { get; set; }  public int CurrentPage { get; set; }  public int TotalPages =>  (int)Math.Ceiling((decimal)TotalItems / ItemsPerPage);  }  } |

Musimy przekazać obiekt klasy **PagingInfo** do widoku, dołączamy go więc do modelu widoku listy ogłoszeń.

|  |
| --- |
| namespace MyAdvert.Core.ViewModels  {  public class AdvertsViewModel  {  public IEnumerable<Advert> Adverts { get; set; }  public IEnumerable<Category> Categories { get; set; }  public FilterAdverts FilterAdverts { get; set; }  public PagingInfo PagingInfo { get; set; }  }  } |