

Mariusz Wieczorek

mariusz.wieczorek@kabat.pl

Streszczenie

Dodanie nowych właściwości do klasy użytkownika  
Utworzenie klasy **ApplicationUser** dziedziczącej po klasie **IdentityUser**

i podmiana **IdentityUser** nowo utworzoną klasą

asp.net Core

IdentityUser -> ApplicationUser

**ApplicationUser**

|  |
| --- |
| Do klasy z użytkownikiem za bardzo nie mamy dostępu, dlatego aby dodać nowe pola do użytkownika musimy utworzyć nową klasę **ApplicationUser**, która będzie dziedziczyła po klasie IdentityUser i w niej dodajemy nowe pola . |

|  |
| --- |
| using Microsoft.AspNetCore.Identity;  using System;  using System.Collections.Generic;  using System.Collections.ObjectModel;  using System.Linq;  using System.Threading.Tasks;  namespace MyTasks.Core.Models.Domains  {  public class ApplicationUser : IdentityUser  {  **// Tutaj dodajemy nowe pola**    [Required]  [Display(Name = "Imię")]  public string FirstName { get; set; }  [Required]  [Display(Name = "Nazwisko")]  public string LastName { get; set; }  [Required]  [Display(Name = "Stanowisko")]  public string Possition { get; set; }  }  } |

Aby klasę **IdentityUser** zastąpić klasą **ApplicationUser**, to musimy w aplikacji poprawić odwołania tam gdzie mamy już odwołania do IdentityUser.

**Musimy te odwołania zmienić w trzech plikach:**

**1. ApplicationDbContext**

Najpierw klasa **ApplicationDbContext** dziedziczy ona po **IdentityDbContext**,  
musimy jeszcze mu dodatkowo przekazać nowo utworzoną klasę **<ApplicationUser>**robimy to w ten sposób:

|  |
| --- |
| namespace MyTasks.Persistence  {  public class ApplicationDbContext : IdentityDbContext  {  public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)  : base(options)  {  }  }  } |
| namespace MyTasks.Persistence  {  public class ApplicationDbContext : IdentityDbContext<ApplicationUser>  {  public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)  : base(options)  {  }  }  } |

**2. Startup.cs**

Drugim miejscem jest klasa **Startup.cs** tam gdzie konfigurujemy serwisy musimy również  
zmienić **IdentityUser** na **ApplicationUser**.

|  |
| --- |
| public void ConfigureServices(IServiceCollection services)  {  services.AddDbContext<ApplicationDbContext>(options =>  options.UseSqlServer(  Configuration.GetConnectionString("DefaultConnection")));  services.AddDatabaseDeveloperPageExceptionFilter();  services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)  .AddEntityFrameworkStores<ApplicationDbContext>();  services.AddControllersWithViews();  } |
| public void ConfigureServices(IServiceCollection services)  {  services.AddDbContext<ApplicationDbContext>(options =>  options.UseSqlServer(  Configuration.GetConnectionString("DefaultConnection")));  services.AddDatabaseDeveloperPageExceptionFilter();  services.AddDefaultIdentity<ApplicationUser>(options => options.SignIn.RequireConfirmedAccount = true)  .AddEntityFrameworkStores<ApplicationDbContext>();  services.AddControllersWithViews();  } |

**3. \_LoginPartial.cshtml**

Trzecim miejscem jest widok częściowy **\_LoginPartial.cshtml**

|  |
| --- |
| @using Microsoft.AspNetCore.Identity  @inject SignInManager<IdentityUser> SignInManager  @inject UserManager<IdentityUser> UserManager |
| @using Microsoft.AspNetCore.Identity  @using MyTasks.Core.Models.Domains  @inject SignInManager<ApplicationUser> SignInManager  @inject UserManager<ApplicationUser> UserManager |

Tworzymy konto, potwierdzamy I logujemy się do aplikacji

