

Mariusz Wieczorek

mariusz.wieczorek@kabat.pl

Streszczenie

Konwencje w Entity Framework Core  
Modyfikacja domyślnych zachowań framework’a za pomocą  
Data Annotations i FluentAPI

Entity Framework Core

Konfiguracja przy użycie FluentApi w odrębnych plikach

# **Konfiguracja encji w odrębnych plikach**

Poniżej przykładowa konfiguracja encji za pomocą **FluentApi** wewnątrz metody OnModelCreating w klasie ApplicationDbContext.

|  |
| --- |
| protected override void OnModelCreating(ModelBuilder modelBuilder)  {  base.OnModelCreating(modelBuilder);  modelBuilder.Entity<UserArea>()  .HasKey(sc => new { sc.AreaId, sc.UserId });    modelBuilder.Entity<UserArea>()  .HasKey(x => x.Id);  modelBuilder.Entity<UserArea>()  .HasOne<ApplicationUser>(sc => sc.User)  .WithMany(s => s.UserAreas)  .HasForeignKey(sc => sc.UserId);  modelBuilder.Entity<UserArea>()  .HasOne<Area>(sc => sc.Area)  .WithMany(s => s.UserAreas)  .HasForeignKey(sc => sc.AreaId);  } |

Możemy przenieś konfigurację do osobnej klasy, a właściwie do osobnych klas.  
Każda encja będzie miała konfigurację w osobnym pliku.  
ApplicationDbContext.cs dołączenie konfigurację z innego pliku, zamiast umieszczania całej konfiguracji w metodzie OnModelCreating

|  |
| --- |
| Protected override void OnModelCreating(ModelBuilder modelBuilder)  {  base.OnModelCreating(modelBuilder);  modelBuilder.ApplyConfiguration(new UserAreaConfiguration());  } |

Klasa konfigurująca encję

|  |
| --- |
| namespace MwMaintenance.Persistence.Configurations  {  public class UserAreaConfiguration : IEntityTypeConfiguration<UserArea>  {  public void Configure(EntityTypeBuilder<UserArea> builder)  {  builder.HasKey(x => x.Id);  //builder.HasKey(sc => new { sc.AreaId, sc.UserId });  builder.HasOne<ApplicationUser>(sc => sc.User)  .WithMany(s => s.UserAreas)  .HasForeignKey(sc => sc.UserId);  builder.HasOne<Area>(sc => sc.Area)  .WithMany(s => s.UserAreas)  .HasForeignKey(sc => sc.AreaId);  }  }  } |

W EF6 wygląda to inaczej niż w EFCore.

|  |
| --- |
| protected override void OnModelCreating(ModelBuilder modelBuilder)  {  base.OnModelCreating(modelBuilder);  // EFCore  modelBuilder.ApplyConfiguration(new UserAreaConfiguration());  // EF6  modelBuilder.Configurations.Add(new UserAreaConfiguration()); |

W sama klasa konfigurująca

|  |
| --- |
| namespace Diary.Models.Configurations  {  public class StudentConfiguration : EntityTypeConfiguration<Student>  {  public StudentConfiguration()  {  ToTable("dbo.Students");  HasKey(x => x.Id);  Property(x => x.FirstName)  .HasMaxLength(100)  .IsRequired();  Property(x => x.LastName)  .HasMaxLength(100)  .IsRequired();  }  }  } |