

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش پیکربندی کلاس های Domain در Code-First

مدرس: مهندس افشین رفوآ

آموزش پیکربندی کلاس های Domain در Code-First

همانطور که می دانید، شما می توانید مدل **entity data** را از یک پایگاه داده موجود در **Database-First** ایجاد کنید. این **EDM** همه اطلاعات اصلی را در **SSDL**، **CSDL** و **MSL** را دارد. بنابراین **EF** می تواند از این مدل در **query** ها، ردیابی تغییرات، قابلیت های بهنگام سازی و غیره استفاده نماید. به همین روش **entity** **Code-First framework** به شما این اجازه را می دهد که با استفاده از کلاس های **Domain** تان، مدل را بسازید که در عوض برای فعالیت دیگری توسط **EF** استفاده می شود. **Code-First** قراردادهای مطمئنی را برای کلاس های **Domain** تان پیشنهاد می کند، بنابراین **EF** می تواند آن را تفسیر نموده و مدل را بسازد. هرچند اگر کلاس های **Domain** تان این قراردادها را نداشته باشد در آن صورت شما می توانید کلاس های **Domain** تان را پیکر بندی کنید بنابراین **EF** می تواند آن را بفهمد و مدل را از روی آن بسازد. دو روش وجود دارد که شما می توانید کلاس های **domain** تان را پیکر بندی کنید.

1. DataAnnotation
2. Fluent API

DataAnnotation

DataAnnotation ویژگی ساده ای که بر اساس پیکربندی می باشد که بر روی کلاس های **Domain** و خصوصیاتش قابل اعمال هستند. شما می توانید اکثر این ویژگی ها را در **namespace** به طریق **System.ComponentModel.DataAnnotations** بیابید. بنابراین **DataAnnotation** زیر مجموعه ای از پیکربندی های **Fluent API** را فراهم می کند. بنابراین اگر برخی از ویژگی ها را نیافتید، باید برای پیکربندی آن از **Fluent API** استفاده کنید.

در پایین مثالی از **DataAnnotation** آورده شده که در کلاس **Student** استفاده شده است.

```
[Table ("StudentInfo")]
public class Student
{
    public Student() { }
    [Key]
```

```

public int SID { get; set; }
[Column("Name", TypeName="ntext")]
[MaxLength(20)]
public string StudentName { get; set; }
[NotMapped]
public int? Age { get; set; }
public int StdId { get; set; }
[ForeignKey("StdId")]
public virtual Standard Standard { get; set; }
}

```

پیکربندی **Fluent API** زمانی اعمال می شود که **EF** مدل را از روی کلاس های **Domain** تان می سازد. شما می توانید پیکربندی ها را به وسیله دوباره نوشتن روی ویژگی **OnModelCreating** کلاس **DbContext** به شرح زیر اعمال کنید.

```

public class SchoolDbContext: DbContext
{
    public SchoolDbContext():
base("SchoolDBConnectionString")
    {
    }
    public DbSet Students { get; set; }
    public DbSet Standards { get; set; }
    public DbSet StudentAddress { get; set; }
    protected override void
OnModelCreating(DbModelBuilder modelBuilder)
    {
        //Configure domain classes using Fluent API
here
        base.OnModelCreating(modelBuilder);
    }
}

```

شما می توانید از **modelBuilder** که شی ای از کلاس **DbModelBuilder** می باشد برای پیکربندی کلاس های **Domain** استفاده کنید.

جزئیات در مورد **DataAnnotation** و **Fluent API** در قسمت بعدی آورده شده است.