

# 第17期-映射关系配置

2019年4月28日 14:16

Microsoft.EntityFrameworkCore.Relational  
using System.ComponentModel.DataAnnotations.Schema;

## 表名设置

默认情况下，表名将使用 DbSet<TEntity> 实体属性名。

```
public DbSet<Blog> Blogs { get; set; }

[Table("blogs", Schema = "blogging")]
public class Blog
{
    public int BlogId { get; set; }
    public string Url { get; set; }
}
```

```
modelBuilder.Entity<Blog>().ToTable("blogs", schema: "blogging");
```

## 列名设置

按照约定，字段名与属性名相同。

```
[Column("blog_id")]
public int BlogId { get; set; }
```

```
modelBuilder.Entity<Blog>().Property(b => b.BlogId).HasColumnName("blog_id");
```

## 数据类型

默认使用与 CLR 兼容的数据库类型，除非有值转换器，针对字段的其它元数据，可手动设置。

### System.ComponentModel.DataAnnotations

```
[MaxLength(10)]
public string Url { get; set; }

[Column(TypeName = "varchar(200)")]
public string Url { get; set; }
[Column(TypeName = "decimal(5, 2)")]
public decimal Rating { get; set; }
```

```

modelBuilder.Entity<Blog>().Property(b=>b.Url).HasColumnType("varchar(200)");
modelBuilder.Entity<Blog>().Property(b => b.Rating).HasColumnType("decimal(5,2)");

modelBuilder.Entity<Blog>(eb =>
{
    eb.Property(b => b.Url).HasColumnType("varchar(200)");
    eb.Property(b => b.Rating).HasColumnType("decimal(5, 2)");
});

```

## 主键设置

按照约定，主键值将使用 PK\_<type name> 约束唯一性。

```

modelBuilder.Entity<Blog>().HasKey(b => b.BlogId).HasName("PrimaryKey_BlogId");

```

## 默认 Schema 架构

按照约定，数据库提供程序将选择最合适的默认架构，SQL server 默认使用 dbo 架构，SQLite 不支持架构。

### Sql Server需要架构(Schema)的真实原因

```

modelBuilder.HasDefaultSchema("blogging");

```

## 计算列设置

按照约定，计算的列不在模型中创建。

```

modelBuilder.Entity<Person>()
    .Property(p => p.DisplayName)
    .HasComputedColumnSql("[LastName] + ', ' + [FirstName]");

```

## 指定默认值

```

modelBuilder.Entity<Blog>().Property(b => b.Rating).HasDefaultValue(3);

```

```

modelBuilder.Entity<Blog>().Property(b => b.Created).HasDefaultValueSql("getdate()");

```