

# 第14期-值转换器

2019年4月26日 07:44

## 值转换器

### 基础知识

值转换器允许向数据库读取或写入数据时，在 模型和数据库之间转换数据。例如：模型字段类型是枚举，保存数据库中的类型为字符串或者INT类型，更新或者插入数据时，需要将枚举转换成数据库类型，查询数据时，需要将数据库类型转换为模型的字段类型。

### 配置值转换器

```
public class Rider
{
    public int Id { get; set; }
    public EquineBeast Mount { get; set; }
}

public enum EquineBeast
{
    Donkey,
    Mule,
    Horse,
    Unicorn
}

modelBuilder
    .Entity<Rider>()
    .Property(e => e.Mount)
    .HasConversion(
        v => v.ToString(),
        v => (EquineBeast)Enum.Parse(typeof(EquineBeast), v));
```

### ValueConverter 值转换器类

```
var converter = new ValueConverter<EquineBeast, string>(
    v => v.ToString(),
    v => (EquineBeast)Enum.Parse(typeof(EquineBeast), v));

modelBuilder
    .Entity<Rider>()
    .Property(e => e.Mount)
    .HasConversion(converter);
```

当多个属性使用相同的转换时，这很有用。

### 微软 EF Core 中内置的转换器

- BoolToZeroOneConverter - Bool to zero and one
- BoolToStringConverter - Bool to strings such as "Y" and "N"
- BoolToTwoValuesConverter - Bool to any two values
- BytesToStringConverter - Byte array to Base64-encoded string
- CastingConverter - Conversions that require only a type cast
- CharToStringConverter - Char to single character string
- DateTimeOffsetToBinaryConverter - DateTimeOffset to binary-encoded 64-bit value
- DateTimeOffsetToBytesConverter - DateTimeOffset to byte array
- DateTimeOffsetToStringConverter - DateTimeOffset to string
- DateTimeToBinaryConverter - DateTime to 64-bit value including DateTimeKind
- DateTimeToStringConverter - DateTime to string
- DateTimeToTicksConverter - DateTime to ticks
- EnumToNumberConverter - Enum to underlying number
- EnumToStringConverter - Enum to string
- GuidToBytesConverter - Guid to byte array
- GuidToStringConverter - Guid to string
- NumberToBytesConverter - Any numerical value to byte array
- NumberToStringConverter - Any numerical value to string
- StringToBytesConverter - String to UTF8 bytes
- TimeSpanToStringConverter - TimeSpan to string
- TimeSpanToTicksConverter - TimeSpan to ticks

```
var converter = new EnumToStringConverter<EquineBeast>();
```

```
modelBuilder
    .Entity<Rider>()
    .Property(e => e.Mount)
    .HasConversion(converter);
```

### 预定义的转换

```
modelBuilder.Entity<Rider>().Property(e => e.Mount).HasConversion<string>();
```

也可通过数据注解的方式指定类型。

```
public class Rider
```

```
{  
    public int Id { get; set; }  
  
    [Column(TypeName = "nvarchar(24)")]  
    public EquineBeast Mount { get; set; }  
}
```

通过接口实现，可定制自己的转换器。