

第07期-导航属性关系配置

2019年4月16日 10:43

多导航属性

在 Post 类中，可能需要跟踪是文章的创建者和最后编辑者，下面是 Post 类的两个新的导航属性。

```
public class Post
{
    public int PostId { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }

    public User Author { get; set; }
    public User Contributor { get; set; }
}

public class User
{
    public string UserId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }

    [InverseProperty("Author")]
    public List<Post> AuthoredPosts { get; set; }

    [InverseProperty("Contributor")]
    public List<Post> ContributedToPosts { get; set; }
}
```

注意：同样可使用 ForeignKeyAttribute 特性显式指定外键名。

```
modelBuilder.Entity<Post>().HasOne(p => p.Author).WithMany(u => u.AuthoredPosts).HasForeignKey("AuthorId");
```

```
modelBuilder.Entity<Post>().HasOne(p => p.Contributor).WithMany(u => u.ContributedToPosts);
```

若要使用 Fluent API 配置关系，您首先确定构成关系的导航属性，使用 HasOne 或 HasMany 开始配置当前实体类型上的导航属性，然后调用 WithOne 或 WithMany 来配置反导航属性。HasOne 和 WithOne 用于引用导航属性，HasMany 和 WithMany 用于集合导航属性。

```
modelBuilder.Entity<Post>().HasOne(p => p.Blog).WithMany(b => b.Posts);
modelBuilder.Entity<Blog>().HasMany(p => p.Posts).WithOne(b => b.Blog);
```

单一导航属性

```
modelBuilder.Entity<Blog>().HasMany(b => b.Posts).WithOne();
```

其中：WithOne 不指定参数即可

显式外键配置

```
referenceCollectionBuilder.HasForeignKey(p => p.BlogForeignKey);
```

```
modelBuilder.Entity<Car>().HasKey(c => new { c.State, c.LicensePlate });
```

```
referenceCollectionBuilder.HasFoHasForeignKey(s => new { s.CarState, s.CarLicensePlate })
```

```
referenceCollectionBuilder.HasForeignKey("BlogId");
```