

## 第08期-实体模型映射与数据库表迁移

2021年1月13日 22:36

### 模型映射

```
public class ProjectEntityTypeConfiguration :
    IEntityTypeConfiguration<Project>
{
    public void Configure(EntityTypeBuilder<Project> builder)
    {
        builder.ToTable("Projects",
            Constants.DbConstants.DefaultTableSchema);
        builder.HasKey(e => e.Id);
        builder.Property(e => e.Name).IsRequired().HasMaxLength(20);
    }
}

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.ApplyConfigurationsFromAssembly(Assembly.GetExecutingAssembly());
    base.OnModelCreating(modelBuilder);
}
```

### 自定义仓储

```
public interface IOrderRepository : IRepository<Order>
{
    Order Add(Order order);

    void Update(Order order);

    Task<Order> GetAsync(Guid orderId);
}

public class OrderRepository : EfCoreRepository<DeviceCenterDbContext,
    Order>, IOrderRepository
{
    public OrderRepository(DeviceCenterDbContext dbContext) :
        base(dbContext) { }

    public Order Add(Order order)
    {
        return DbSet.Add(order).Entity;
    }

    public async Task<Order> GetAsync(Guid orderId)
    {
        var order = await DbSet.Include(x =>
            x.Address).FirstOrDefaultAsync(o => o.Id == orderId);

        if (order == null)
```

```

        {
            order = DbSet.Local.FirstOrDefault(o => o.Id == orderId);
        }

        if (order != null)
        {
            await _dbContext.Entry(order).Collection(i =>
i.OrderItems).LoadAsync();
            await _dbContext.Entry(order).Reference(i =>
i.OrderStatus).LoadAsync();
        }

        return order!;
    }

    public void Update(Order order)
    {
        _dbContext.Entry(order).State = EntityState.Modified;
    }
}

```