**Order Service:**

public class OrderService : BaseService<Order>, IBaseService<Order>

{

public OrderService(IServiceProvider provider)

:base(provider)

{

RuleFor(t => t.Date).CustomValue(t => t == null, "Please specify the order date");

RuleFor(t => t.OrderDeatils).CustomValue(t => t == null || !t.Any(), "The order must have at least one product");

RuleForEach(t => t.OrderDeatils).SetValidator(new OrderDeatilService(provider));

RuleFor(t => t.OrderStatus).Custom(t => t.DeliveryId.HasValue && t.OrderStatus == OrderStatus.Canceled,

"The order has a courier and it is not possible to cancel it.");

}

public override Task<Order> AddAsync(Order entity)

{

var maxOrderNo = GetAll().Where(t => t.Date == entity.Date).Max(t => (int?)t.OrderNo).GetValueOrDefault() + 1;

entity.OrderNo = maxOrderNo;

return base.AddAsync(entity);

}

public override async Task UpdateAsync(Order entity)

{

var old = await SingleAsync(entity.Id);

entity.OrderNo = old.OrderNo;

await base.UpdateAsync(entity);

}

}