## 一、操作日志

1、Log表结构设计

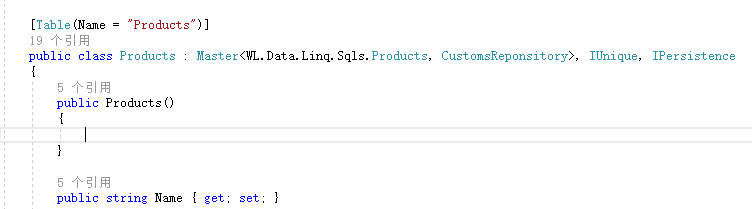
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 描述 | 类型 | 是否可空 |
| ID | 主键 | nvarchar(50) | 否 |
| Name | 对象名称，  例如：Clients,Orders | nvarchar(50) | 否 |
| MainID | 对象ID,  例如：OrderID | nvarchar(50) | 否 |
| AdminID | 操作人ID,  例如：UserID、AdminID  会员等保存UserID | nvarchar(50) | 否 |
| Summary | 操作内容 | nvarchar(MAX) | 否 |
| Json | 操作数据 | nvarchar(MAX) | 否 |
| CreateDate | 创建时间 | datetime | 否 |

### 操作内容

需要明确：人 时间 操作内容。

界面显示：按时间倒序排序显示。

### 代码说明



1）、在需要操作日志的对象中添加自定义标签属性。

2）、通过对象的Log拓展函数持久化日志内容。

var adminID = "201900100901";

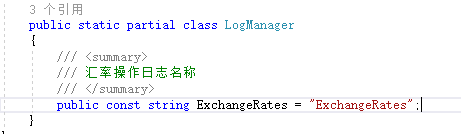
var adminName = "黄云";

Products product = new Products();

product.ID = Guid.NewGuid().ToString();

product.Log(adminID, $"管理员[{adminName}]完成了封箱");

1. 、日志的查询



配置日志的名称。

视图调用：

LogsView view = Needs.Wl.Services.LogManager.Logs[Needs.Wl.Services.LogManager.ExchangeRates];

view.AllowPaging = false;//根据自己的业务需求，设置是否分页

var list = view.ToList();

## 异常日志的记录

1、ExceptionLogs表结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 描述 | 类型 | 是否可空 |
| ID | 主键 | nvarchar(50) | 否 |
| Source | 导致错误的应用程序或对象的名称 | nvarchar(200) | 否 |
| StackTrace | 调用堆栈 | nvarchar(max) | 否 |
| Message | 异常的消息 | nvarchar(max) | 否 |
| CreateDate | 创建时间 | datetime | 否 |

try

{

...

}

catch (Exception ex)

{

ex.Log();

}

在catch中捕获异常后，调用Exception 类型的拓展函数Log(),持久化日志。