AyaORM数据交互框架设想

S

目录

[标题 1](#_Toc2618390)

[目录 2](#_Toc2618391)

[修改记录 3](#_Toc2618392)

[1、 标题1 4](#_Toc2618393)

[1.1、 标题2 4](#_Toc2618394)

[1.1.1、 标题3 4](#_Toc2618395)

修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 作者 | 说明 | 版本号 |
|  |  |  |  |

# 应用情况

## 说明

以下例子都与数据交互有关，设计的时候要考虑以下这些应用情况/场景，怎样设计最方便。

## 应用场景-例子

### 博客文章管理

会用到分页查询（列表查询），文章浏览量控制，文章评论控制，文章富文本等逻辑处理。

### 图标统计分析

图表统计分析，用的数据通常比较独特，受制于其他框架的影响（如echats），会要求返回的数据格式以及结构。

这种情况，框架肯定不能直接返回你想要的数据，都需要查询出数据表list，然后再对数据进行细节处理。

面对这种情况，我们的框架能不能尽量减少这种数据细节处理代码量，比如写一个ChartsService类，继承我们的数据基础处理类，利用框架的方法组合，来完成数据处理。

# 怎样设计调用方式比较好

## 设想

以下设想其实可以结合起来，让我们的orm全部支持，但是要想好代码架构怎样能同时实现，并且不会增加太多复杂逻辑。

### linq链式调用

List<T> model = db.From(“TableName”)

.Where(“name=@name”)   
.Select<T>(“columns field”);

### 简单封装Dapper，简单调用

T model db.GetList<T>(new {name=name});

# 肯定会需要的类

## SqlBuilder：sql语句生成器

用来构造sql（SqlStatement对象）。

## SqlStatement：sql语句实体类

类似于SqlCommond的感觉。

# 根据使用情况设计使用模式

## 建造者模式

### 说明

需要先实现一个包括增删改查等基础功能的抽象DAO类。

DAO 要怎样设计，直接包装一层dapper+sqlbuilder 带来的坏处是什么?

dao让具体的业务Service来继承并扩展，通过Service注入来使用，也可以自由实例化调用，new ArticleService().Get()。