# Alxf框架说明

全文说明:  
DM=DataModel 数据模型  
BM=BusinessModel 业务模型  
特性:修饰在类,或其方法,属性上的Attribute  
客户端:在大部分情况下,是指aspx.cs之类的类库消费端

## 总体说明

总体思路:

1. 构建简单的页面调用和验证程序,
2. 使用~~linq~~ Entity Framework访问数据库
3. 在
4. 权限管理使用membership,页面级的权限控制在模板页中结合Web.sitemap进行筛选
5. 服务器端的验证模型
6. 客户端js的验证模型:根据业务模型的特性翻译成简单的js,供jquery.alxf.validate验证框架统一调用

Frame下的框架的主体代码,并不是所有的都有用

此框架适用于有些规模的程序,小项目并不适合,小项目的话

功能:业务模型的操作

~~分层问题:DAL可以不用,全部放到BLL中~~

分离用户管理到其他的库(注意区别用户和人员)

Alxf.DAL

从数据库的角度划分文件夹,

缺点,不同文件夹的代码理论上不穿插,

Partial文件夹的放linq to sql的分布类,主要实现默认值等保证数据库不会报异常的简单功能

Alxf.BLL

从DM和BM的互转,并做业务上验证

每个业务模型实现接口

虽然业务模型和数据模型不是一一对应(废话,如果是还要这层做啥),但业务模型一定会有一个主数据模型对应,如果确实没有,数据拆的很散,就在DAL写一个数据级的视图(不推荐)

## 代码生成

使用~/DevTool/creatClass.aspx来根据数据库类 生成业务类,由于2者几乎不可能有可遵循的关系,所以此功能只是一个简单的帮助工具

生成的业务模型 和 数据模型的转换函数

1使用~/DevTool/AddForm.aspx来生成代码

2 验证在linq to sql 的model分布类文件使用字段的OnXXChanged()方法上班使用Validate,框架会根据attr来翻译成js来进行验证,写特性的时候一定要吧4个属性全部赋值,不能有null

3非model类(比如:查询条件)

查询条件的收集,通过xml进行配置,使用通用条件收集类

4客户端验证:

单一验证: 字段长度,类型,范围------使用attr进行翻译,只保证数据类型正确

翻译BM的特性,翻译成js片段,~~链接~~注入到页面中

Js验证调用框架使用链接的方式,验证规则直接输入到页面中

问题:服务器端的BM属性如何与页面上要验证的表单元素对用

联合验证: 密码重复的是否正确

Ajax验证:用户名重复性验证

验证合格后马上锁定,要双击解锁后才能重新输入,提交时保证是锁定的

5布局:

用ul控制的 单行 或单列的 布局样式

6 用户控件

尽量不涉及数据访问,这里主要封装的是 业务化的 UI和js

对于有上传需求的页面,把上传功能单独做一个页面,不要混杂在表单填写中

## 3设计原则与约定

数据库:简化数据类型 int float guid 字符 时间

放弃char(1), 使用int+枚举替代

尽力不用null字段,目前能想到的时间类型可以为null的是利大于弊的

考察字典表的修改可能,考虑3中方式:

1. 枚举:在程序中写死,不可修改
2. Xml:和用户业务关系不大,很少修改,
3. DB:使用2张通用的枚举存储表

如果用户的新功能和原来某功能类似,则修改原来功能(以增加为主),培训用户多点几次鼠标

设计最贴近用户的实体类,去除外键的概念,添加查询等操作一律使用text, 对text到val的转换进行封装(注意缓存),

## 4Demo功能

### 视频讨论功能:

目前只有动画的讨论功能

动画

### 流水账:

功能:

收入支出记账,收入支出分类,统计

个人门户?

微博?