关于Lind.DDD.Manager的培训与学习

讲解：张占岭

主要框架：Lind.DDD,Lind.DDD.Manager

目录

[关于Lind.DDD.Manager 2](#_Toc19850)

[如何为你的项目安装Lind.DDD.Manager 2](#_Toc17084)

[关于几大过虑器，代替配置麻烦和过时的HttpModule 3](#_Toc4731)

[关于控制器与模块 3](#_Toc18603)

[关于角色－菜单－操作权限的设计 4](#_Toc23426)

[关于Action中添加权限特性标识 4](#_Toc20919)

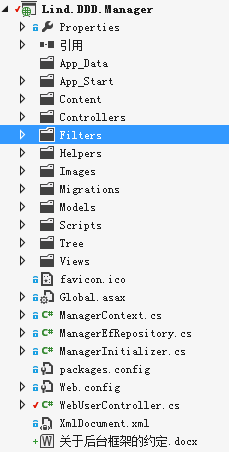
[关于Action中添加操作日志特性标识 5](#_Toc13848)

[关于数据集权限 6](#_Toc31336)

[关于菜单权限的安全设计 7](#_Toc17869)

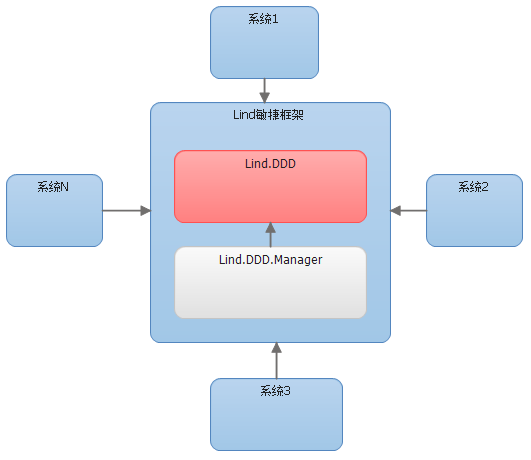
### 关于Lind.DDD.Manager

由于数据模型，数据库初始化（Code.First自动升级数据库或者进行数据库版本的迁移）控制器，View视图，css,js等元素组件的一套**标准的后台管理系统**框架，可以直接应用到任何一个系统上，可以它将发布到Nuget上，以后安装和更新更加方便。



### 如何为你的项目安装Lind.DDD.Manager

Lind在nuget上早已有了自己的版本，用户在自己的项目上直接从nuget上在线安装即可，它将会把对应的view,content,script,dll等安装到你的对应项目中，你不必担心它的升级，因为一切由nuget来控制！



### 关于几大过虑器，代替配置麻烦和过时的HttpModule

Lind.DDD.Authorization.Mvc.AuthorizationLoginFilter：对用户登陆后的授权

Lind.DDD.Filters.ActionAuthorityAttribute：对指定Action添加操作权限的标识

Lind.DDD.Filters.ActionLoggerAttribute：对指定Action进行操作日志的记录

Lind.DDD.Filters.ActionTrackAttribute:对action执行时间的跟踪

Lind.DDD.Filters.ExceptionErrorLoggerAttribute:对action执行过程中，异常的捕获

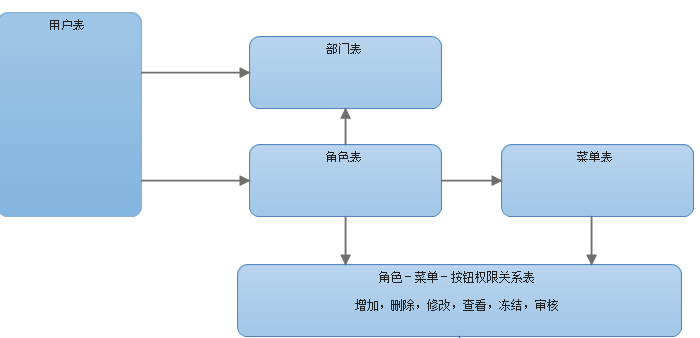
Lind.DDD.Manager.Filters.ManagerUrlAttribute:对指定Url与系统Url进行对比，用于菜单权限的控制

Lind.DDD.Manager.Filters.ManagerActionLoggerAttribute:后台操作日志记录，它继承自ActionLoggerAttribute类，主要实现了后台日志持久化机制

### 关于控制器与模块

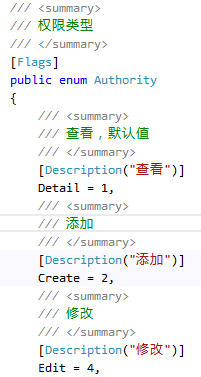
后台模块设计，要求每个模块属于一个controller(控制器)，这样可以更好控制每个action的行为权限。

### 关于角色－菜单－操作权限的设计

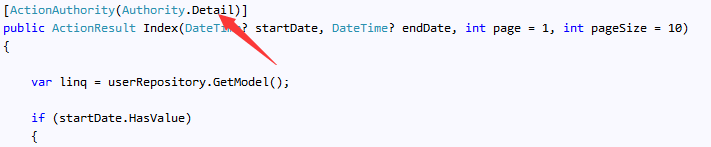


### 关于Action中添加权限特性标识

行为类型的枚举



为每个Action应用自己的行为特性



### 关于Action中添加操作日志特性标识

由于自己的业务系统在记录日志时的持久化方式比较多，所以我们将获取日志的逻辑与持久化的逻辑进行分离，获取放在Lind.DDD里，而持久化的逻辑放在具体的项目中，如Lind.DDD.Manager项目，它有自己的操作日志过滤器，它项目里的Action应用它的ManagerActionLoggerAttribute即可，代码如下



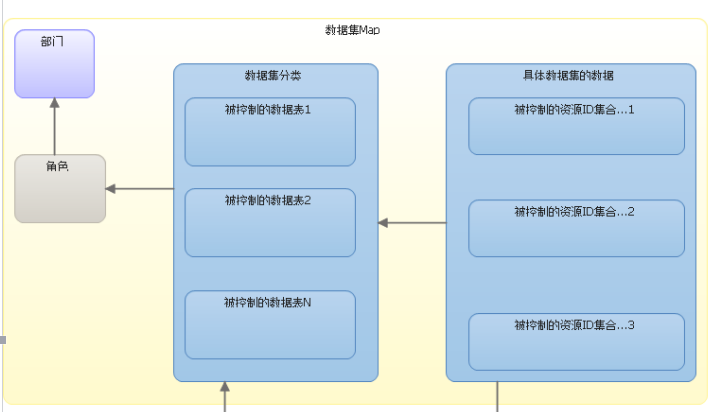
在Action上应用代码如下



### 关于数据集权限

对业务平台的相关数据表的某些字段进行控制，根据不同的角色，为业务平台返回不同的数据集合，而对于业务平台来说，它们将成为对表进行过滤的一种条件！

模型的设计



先添加数据集的类型，用于从业务平台选择需要筛选的所有数据



然后添加对应的值，选对应的类型，可以被筛选出来的目标数据



业务平台可以根据自己的用户编号userid,表名type和字段名field来获取指的数据，框架中提供了/WebDataSetting/GetDataSet方法，返回为目标值组成的字符串，每个值用逗号做分割。

### 关于菜单权限的安全设计

在Lind.DDD.Manager里，菜单权限会被分配到每个角色中，而用户登陆后，会根据它的角色来显示对应的菜单列表，这通常表现在左侧的导航菜单上，而如果用户在地址栏直接输入菜单的URL进行访问时，我们也有对应的ManagerUrlAttribute对它进行控制，主要逻辑是使用数据库里的菜单与当前角色所拥有的菜单进行比对，如果存在这个权限，菜单对应的页面才可以被访问。

