

目录

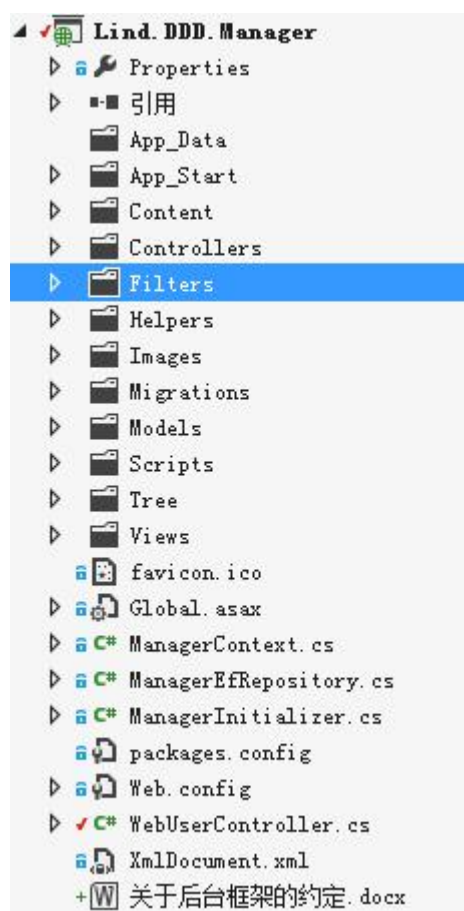
关于 Lind.DDD.Manager.....	2
如何为你的项目安装 Lind.DDD.Manager.....	2
关于几大过滤器，代替配置麻烦和过时的 HttpModule.....	3
关于控制器与模块.....	4
关于角色 - 菜单 - 操作权限的设计.....	4
关于 Action 中添加权限特性标识.....	4
关于 Action 中添加操作日志特性标识.....	5
关于数据集权限.....	6
关于菜单权限的安全设计.....	7

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。

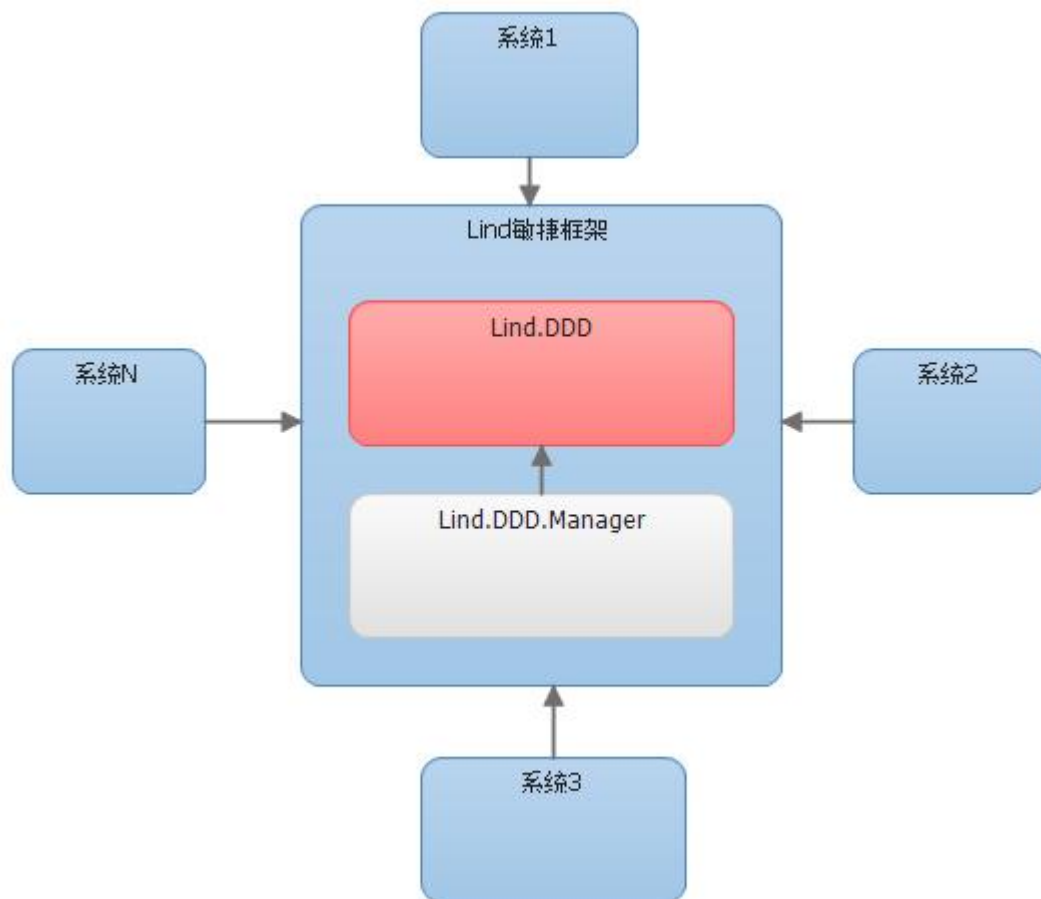


如何为你的项目安装 Lind.DDD.Manager

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 !



关于几大过滤器，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

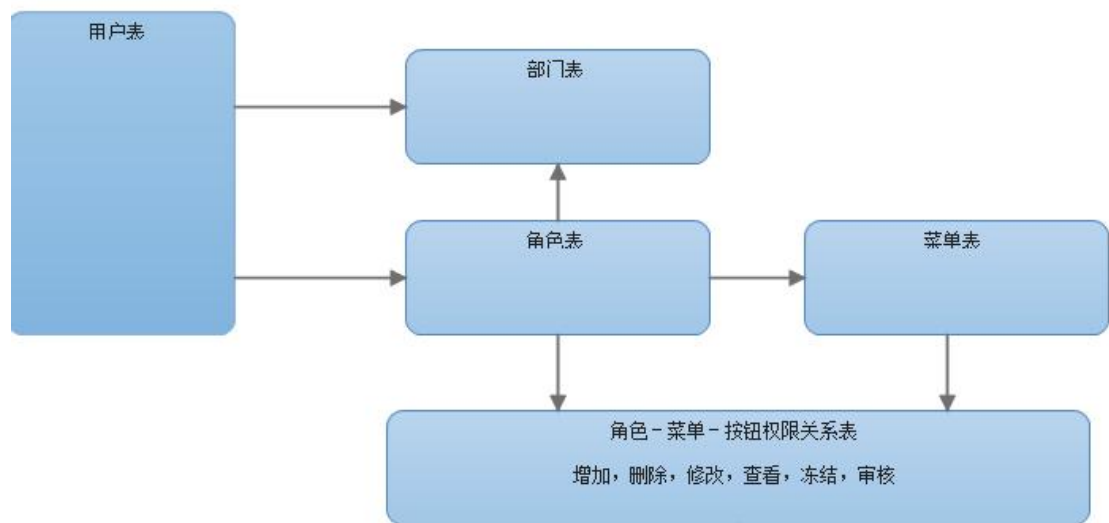
`Lind.DDD.Manager.Filters.ManagerUrlAttribute`:对指定 Url 与系统 Url 进行对比，用于菜单权限的控制

`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

```
/// <summary>
/// 权限类型
/// </summary>
[Flags]
public enum Authority
{
    /// <summary>
    /// 查看, 默认值
    /// </summary>
    [Description("查看")]
    Detail = 1,
    /// <summary>
    /// 添加
    /// </summary>
    [Description("添加")]
    Create = 2,
    /// <summary>
    /// 修改
    /// </summary>
    [Description("修改")]
    Edit = 4,
```

为每个 Action 应用自己的行为特性

```
[ActionAuthority(Authority.Detail)]
public ActionResult Index(DateTime? startDate, DateTime? endDate, int page = 1, int pageSize = 10)
{
    var linq = userRepository.GetModel();

    if (startDate.HasValue)
    {
```

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

```
/// <summary>
/// Action操作日志特性
/// </summary>
public sealed class ManagerActionLoggerAttribute : Lind.DDD.Filters.ActionLoggerAttribute
{
    /// <summary>
    /// 初始化
    /// </summary>
    /// <param name="description">功能说明</param>
    /// <param name="param">说明参数,可以为空</param>
    public ManagerActionLoggerAttribute(string description, params string[] param)
        : base((e) =>
        {
            int userid;
            int.TryParse(CurrentUser.UserID, out userid);
            var db = new ManagerContext();
            var repository = new ManagerEfRepository<WebLogger>(db);
            var webLogger = new WebLogger
            {
                ActionName = e.ActionName,
                ControllerName = e.ControllerName,
                Description = e.Description,
                UserId = userid,
                UserName = CurrentUser.UserName,
                RequestParams = e.RequestParams,
                Authority = e.Authority,
            };
            repository.Insert(webLogger);

            }, description, param) { }
```

```
public override void OnAuthorization(AuthorizationContext filterContext)
{
    base.OnAuthorization(filterContext);
}
```

在 Action 上应用代码如下

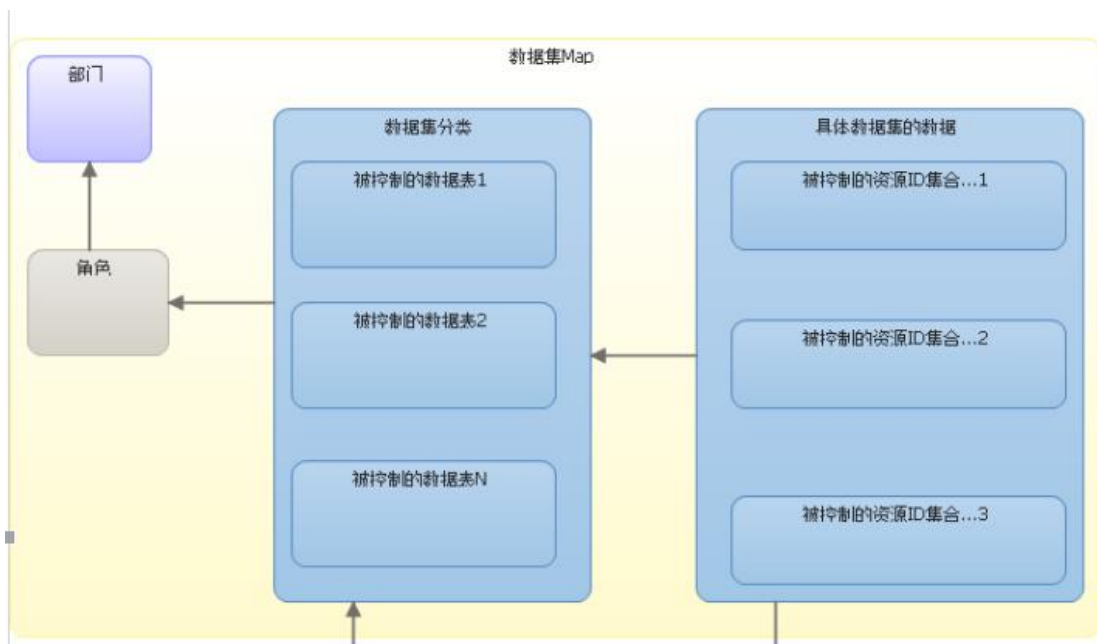
```
[ActionLoggerAttribute("添加用户")]
[ActionAuthority(Authority.Create)]
[HttpPost]
public ActionResult Create(WebManageUsers entity, FormCollection form)
{
    if (GetRole == null || GetDept == null)
    {
        ModelState.AddModelError("", "请把表单填写完整...");
        return View();
    }

    if (userRepository.Find(i => i.LoginName == entity.LoginName) != null)
    {
        ModelState.AddModelError("", "名称不能重复...");
        return View();
    }
}
```

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](#)

显示名称	数据控制的类型	数据控制的条件	数据控制条件需要的数据的API
订单	Order	City	http://localhost:4829/WebDataSetting/GetCity
订单	Order	Price	http://localhost:4829/WebDataSetting/GetPrice

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](#)

角色	类型	类型字段	结果ID集合
管理员	订单	Order->City	1,2
管理员	订单	Order->Price	100,600

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。

