

[知识库](#) [专题](#) [.NET技术](#) [Web前端](#) [软件设计](#) [手机开发](#) [软件工程](#) [程序人生](#) [项目管理](#) [数据库](#) [最新文章](#)您的位置: [知识库](#) » [.NET技术](#)

使用Areas分离ASP.NET MVC项目

作者: [Lyon.L](#) 来源: [博客园](#) 发布时间: 2012-05-31 15:29 阅读: 16558 次 推荐: 14 [原文链接](#) [\[收藏\]](#)

为什么需要分离？

我们知道MVC项目各部分职责比较清晰，相比较ASP.NET Webform而言，MVC项目的业务逻辑和页面展现较好地分离开来，这样的做法有许多优点，比如可测试，易扩展等等。但是在实际的开发中，随着项目规模的不断扩大，Controller控制器也随之不断增多。如果在Controllers文件夹下面有超过两位数controller，即便采用良好的命名规范，或者用子文件夹的形式区分不同功能的控制器，还是会影响项目的可读性和可维护性。因此，在一些场景下，如果能将与某功能相关的文件分离到一个独立的项目中是非常有用的。Asp.Net MVC提供了Areas(区域)的概念达到这一目的。

一个典型的场景

Web应用通常会有前台（面向用户）和后台（面向管理员）两部分，我们希望以/localhost/Admin开始的URL都为后台管理地址，因此我们也许会有这样的路由：

```
routes.MapRoute(    //Admin Route
    "Admin", // Route name
    "Admin/{controller}/{action}/{id}", // URL with parameters
    new { controller = "Admin", action = "Index", id = UrlParameter.Optional } // Parameter defaults
);

routes.MapRoute(    //Default Route
    "Default", // Route name
    "{controller}/{action}/{id}", // URL with parameters
    new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter defaults
);
```

运行程序访问localhost/Admin，通过RouteDebugger（[Asp.Net MVC路由调试的好帮手RouteDebugger](#)）的输出信息可以看到，第一个路由(Admin)匹配成功，AdminController的Index方法被调用了，然而再仔细想想，通过/localhost/Admin/Index可以匹配第二个路由(Default)，同样可以调用AdminController的Index方法！以此类推，后台用户管理列表(/localhost/Admin/User/List)等同于(/localhost/User/List)。

第一次改进

我们要达到一种目的，那就是/localhost/Admin/Admin/Index正确匹配第一个路由，而/localhost/Admin/Index不匹配第二个路由。因此可以对Default路由进行条件限制，参考Stephen Walther的ASP.NET MVC Tip #30 – Create Custom Route Constraints，修改Default路由为：

```
routes.MapRoute(    //Default Route
    "Default", // Route name
    "{controller}/{action}/{id}", // URL with parameters
    new { controller = "Home", action = "Index", id = UrlParameter.Optional }, // Parameter defaults
    new { controller = new NotEqual("Admin")}
);
```

上面的路由意思是AdminController不会匹配Default路由。现在分别运行之前的两个URL，会发现直

 搜索文章

推荐链接

[程序员找工作，就在博客园招聘频道](#)

[程序员问答平台，解决您的技术难题](#)

.NET技术热门文章

[两分钟彻底让你明白Android Activity生命周期\(图](#)

[memcached完全剖析-1. memcached的基础](#)

[经典讲解C# get set](#)

[C#多线程学习\(二\) 如何操纵一个线程](#)

[C#多线程学习\(一\) 多线程的相关概念](#)

.NET技术最新文章

[.NET技术+25台服务器怎样支撑世界第54大网站](#)

[OWIN初探](#)

[使用缓存的9大误区（下）](#)

[使用缓存的9大误区（上）](#)

[项目代码风格要求](#)

最新新闻

[微软正在开发To-do应用Wunderlist后继者Project](#)

[你可能没有发现，Google正悄悄布局离线功能](#)

[创业者需要了解的12项关键绩效指标](#)

[IBM研究人员测量了单个量子点的热传导](#)

[英特尔Atom C2000芯片缺陷会让设备变砖](#)

热门新闻

[秒高铁！中国磁悬浮要飙600公里时速](#)

[任性！刘强东：5年以上员工每人一部iPhone 7](#)

[董明珠再度炮轰：买回的技术都落后，中国人迷信](#)

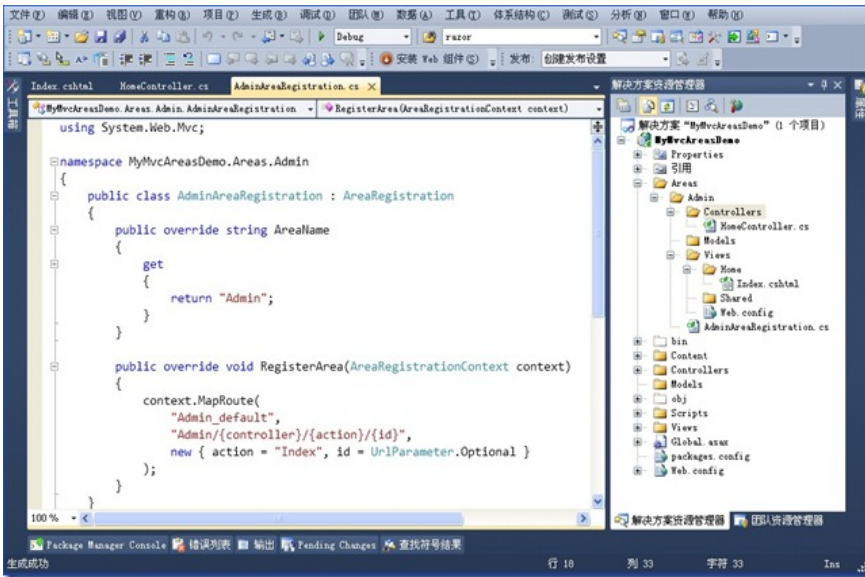
[支付宝种下的树长这样？网友亲见：不愿相信](#)

[微软详解.NET语言战略：C#、VB、F#都不落](#)

访问/Admin/Index已经不能匹配到任何路由。通过修改NotEqual类，可以很容易为Default路由添加多个“排除在外”的Controller限制条件。但是，你不觉得这样很挫么？

第二次改进

这里进入正题，使用Areas这个功能来进行分离。新建一个项目"MyMvcAreasDemo"，然后在项目上右键->添加->Areas，输入"Admin"，如下：



在Areas/Admin/Controllers文件夹下面新建HomeController并添加一个Index的方法和对应的View文件。这里可以发现Areas的另一个好处：你可以在不同Areas下面添加相同名称的Controller。当然，如果你直接这么运行会得到一个错误：



这种情况需要修改一下AdminAreaRegistration.cs和Global.asax，分别为路由加上命名空间限制：

/Areas/Admin/AdminAreaRegistration.cs

```
context.MapRoute(
    "Admin_default",
    "Admin/{controller}/{action}/{id}",
    new { action = "Index", id = UrlParameter.Optional },
    new string[] { "MyMvcAreasDemo.Areas.Admin.Controllers" }
);
```

/Global.asax.cs

```
routes.MapRoute( //Default Route
    "Default", // Route name
    "{controller}/{action}/{id}", // URL with parameters
    new { controller = "Home", action = "Index", id = UrlParameter.Optional }, // Parameter defaults
    new string[] { "MyMvcAreasDemo.Controllers" }
);
```

这样，我们就可以把所有与后台管理相关的Controller和View文件放到Areas/Admin下面，以此类推，可以添加诸如会员(Member)，博客(Blog)，论坛(Forum)等多个Areas。各部分都有自己的顶层文件夹，物理文件都分离开来，管理起来比较方便。

这种方式已经有了很大提高，但是所有的文件还是放在同一个项目里面。当项目规模较大的时候，比较好的开发方式是将不同功能模块按需要独立到不同项目里面，最后再整合成一个整体。这样，每一个项目可以独立开发，测试和发布。

第三次改进

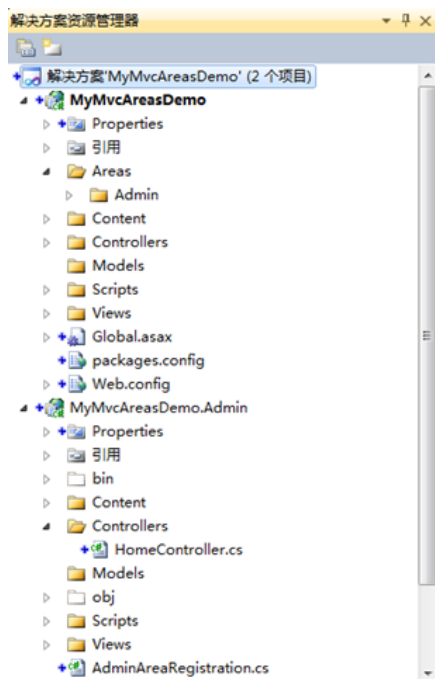
我们需要对现有项目进行一下改造：

1. 在解决方案上面新建一个MyMvcAreasDemo.Admin的MVC3项目，并且删除Global.asax和Web.config两个文件
2. 在根目录新建一个AdminAreaRegistration的类，输入如下内容：

```
public class AdminAreaRegistration : AreaRegistration
{
    public override string AreaName
    {
        get
        {
            return "Admin";
        }
    }

    public override void RegisterArea(AreaRegistrationContext context)
    {
        context.MapRoute(
            "Admin_default",
            "Admin/{controller}/{action}/{id}",
            new { action = "Index", id = UrlParameter.Optional }
        );
    }
}
```

3. 删除MyMvcAreasDemo项目/Areas/Admin文件夹下面的AdminAreaRegistration.cs文件以及Controllers文件夹（包括HomeController）
4. 在MyMvcAreasDemo.Admin项目的Controllers里面新建一个HomeController
5. 记得保留MyMvcAreasDemo/Areas/Admin下面的Views，并且在MyMvcAreasDemo项目里面引用MyMvcAreasDemo.Admin项目，如图：

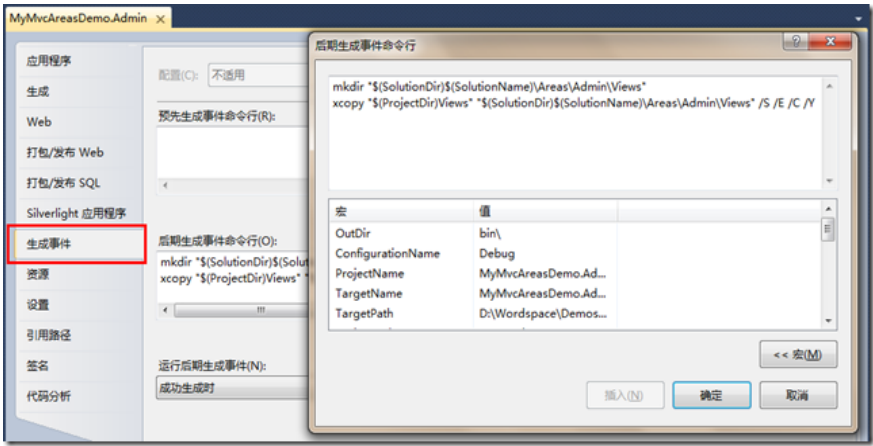


现在运行程序并访问/Admin/Home/Index可以发现Admin项目生效了。这样，我们可以将所有的与后台管理相关的Controller都放到这个新的项目中来。但是很快你会发现，一个新的“问题”又出现了：

当你在MyMvcAreasDemo.Admin里面的HomeController添加新的Action（例如List），然后习惯性在上面右键-"Add View"后，你会发现新增的List.cshtml文件会出现在MyMvcAreasDemo.Admin/Views/Home下面，然后当你访问/Admin/Home/List的时候浏览器会得到错误提示："The view 'List' or its master was not found or no view engine supports the searched locations..."。原来它只会在主程序MyMvcAreasDemo中的对应目录去寻找View。这么一来，MVC框架提供给我们的脚手架功能就大打折扣，当然你可以手动在MyMvcAreasDemo/Areas/Admin/Views中对应添加View，或者在MyMvcAreasDemo.Admin项目中自动生成了View之后再拷贝过去。有没有更好的办法呢？

第四次改进

为了使我们在MyMvcAreasDemo.Admin自动生成的View自动同步到MyMvcAreasDemo/Areas/Admin/Views文件夹中，可以使用“生成事件（Build Event）”里的“Post-Build Event”，打开MyMvcAreasDemo.Admin的属性，修改如下图所示：



我本地的生成事件为：

```
mkdir "$(SolutionDir)$(SolutionName)\Areas\Admin\Views"
xcopy "$(ProjectDir)\Views" "$(SolutionDir)$(SolutionName)\Areas\Admin\Views" /S /E /C /Y
```

意思其实很简单，相信大家都能看得懂，就是完全复制MyMvcAreasDemo.Admin的Views文件夹下所有文件至MyMvcAreasDemo/Areas/Admin/Views中。

现在再次访问/Admin/Home/List就可以得到正确结果了，并且你可以发现List.cshtml已经被复制到MyMvcAreasDemo/Areas/Admin/Views/Home目录里。

至此大功告成！我们已经将面向前台和面向后台的模块成功分离到两个独立的项目中，希望对您有所帮助！

14 0

标签： ASP.NET MVC

» 下一篇： [原创]谈谈WCF中的Data Contract （1）： Data Contract Overview