

.NET Core 微服务 权限系统+工作流

DotNet 今天

(给DotNet加星标，提升.Net技能)

转自：Jade_K
cnblogs.com/wms01/p/10903646.html

一、前言

实际上权限系统老早之前我就在一直开发，大概在刚毕业没多久就想一个人写一个系统，断断续续一直坚持到现在，毕竟自己亲动手自写的系统才有收获，本篇仅介绍权限。小小系统上不了台面，望各位大神勿喷。

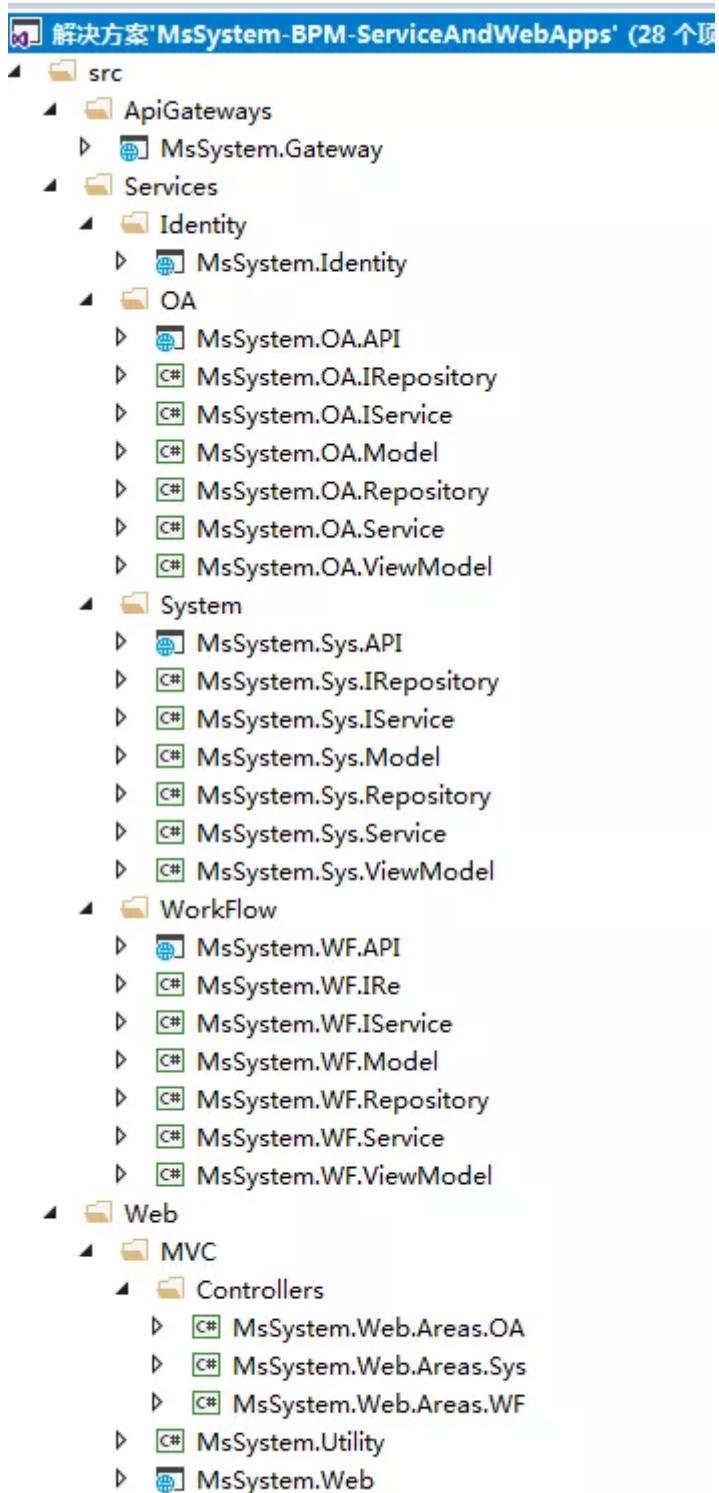
二、系统介绍

目前采用的是.NET Core微服务的方式实现，本文不讨论具体的中间件主要是（ocelot + consul 等），一直参考微软的 eShopOnContainers，进行简单的实现，但是ORM是用的Dapper，并简单进行封装 <https://www.nuget.org/packages/JadeFramework.Dapper/>

当然自己也封装了一些简单的插件进行复用

<https://github.com/wangmaosheng/JadeFramework>

如下：



三、权限系统

权限系统实现很简单，权限的划分我觉得可以分为三种：

1、菜单权限

2、按钮权限

3、数据权限

简单介绍下：

1、菜单权限。表示用户是否能够访问该页面（角色挂钩）

2、按钮权限。表示用户是否能够操作该页面上的功能（角色挂钩）

3、数据权限。表示用户访问页面时进行数据筛选（该功能暂未实现，这个要与具体的业务结合才能写），与部门挂钩，这个不太好理解，当然一般的权限系统这个功能也不会做，举个简单例子，OA系统里面我查看我的工资条，我应该只能看到我自己的数据，但是我的部门经理，他可以有权限看到该部门的全部数据，这个就是数据权限。

为什么写这个系统？

之前待过好几家公司，发现他们的系统都是对菜单进行分配，当然了，业务需求只要这个就当我说，我只是觉得这样做太不安全并且我觉得之前系统的实现方式可以进行一些优化，所以就一直写到现在，可能代码质量不如哪些大神的优秀，系统在我看来太小，就简单搭了个框架实现。你过条小水沟，没必要造条桥。

要使用该系统前提条件：前端：Sea.js和Vue，对于sea.js，在前端这块感觉已经没多少人用了，但是这中CMD思想是不会被淘汰的，你看最近比较火的layerui也是的，对于Vues只是简单的应用，也就用到双向绑定而已，开发复杂的页面确实比较方便，但是简单的页面就得不偿失了。

后端：consul、rabbitmq，具体怎么安装不在描述

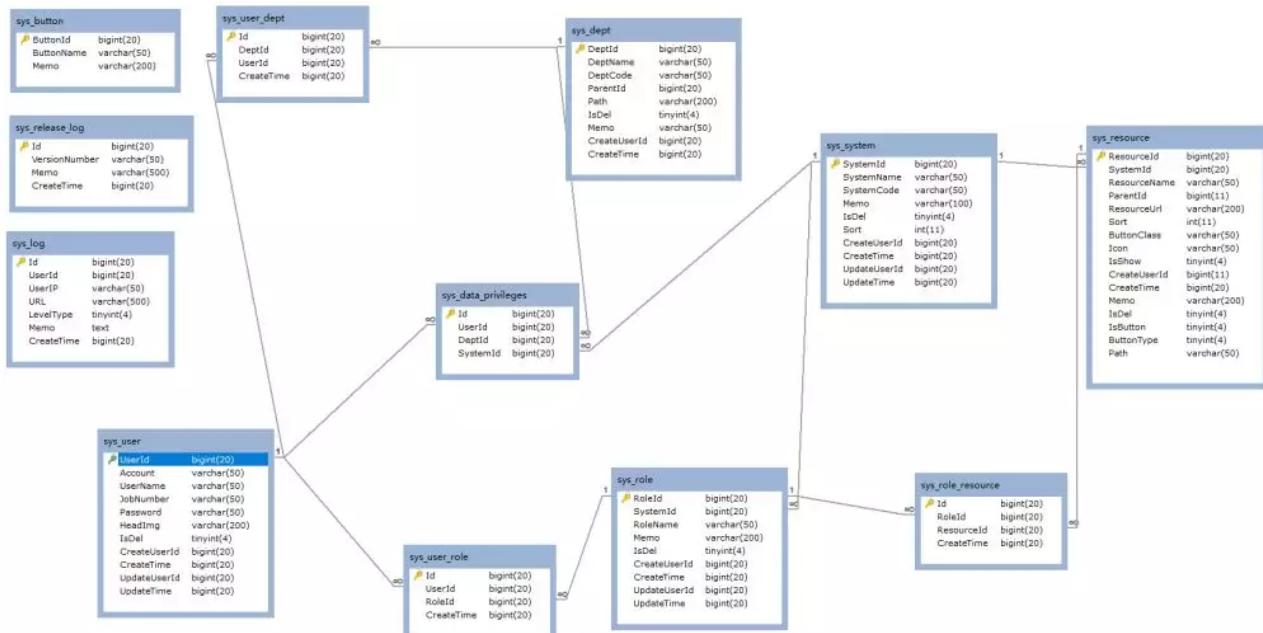
大概的用户访问流程描述如下：

用户登录 ----> 获取该用户角色 ----> 通过角色获取该角色对应的权限 并集 ----> 返回相应数据

sys_user_role

sys_role_resource

系统关系图如下（MySQL）：



具体功能实现请看代码，这里不做阐述，菜单权限的分配通过角色表和菜单表的关联表操作即可，但是按钮的权限分配如何实现？我的实现方式是：把按钮的操作也看成一种菜单的资源分配，只不过比较特殊，我这里不仅仅是对按钮的显示进行控制，我做的比较绝，也对后台方法访问权限也做了控制，这样比较安全，对于按钮权限的控制，实际上是明确的，比方说，一个删除按钮，它只能对应后台的一个删除方法，这个方法是明确的，对于页面的按钮的类型和个数是固定的，不然你没办法分配，基于这个前提，我对菜单的生成进行代码控制从而达到控制目的，因此，菜单和按钮和在一起称之为资源表 `sys_resource`。具体的实现代码也不是很复杂，一层一层判断即可，权限过滤器如下：

```

public class PermissionAuthorizationRequirement : IAuthorizationRequirement
{
    public UrlAndButtonType UrlAndButtonType { get; }

    public PermissionAuthorizationRequirement(string url, ButtonType buttonType)
    {
        UrlAndButtonType = new UrlAndButtonType()
        {
            Url = url,
            ButtonType = (byte)buttonType,
            IsPage = isPage
        };
    }

    public PermissionAuthorizationRequirement(string url, byte buttonType, bool isPage)
    {
        UrlAndButtonType = new UrlAndButtonType()
        {
            Url = url,
        };
    }
}

```

```
        ButtonType = buttonType,
        IsPage = isPage
    };
}

/// <summary>
/// 权限过滤器
/// </summary>
[Authorize]
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class PermissionAttribute : TypeFilterAttribute
{
    /// <summary>
    /// 构造器
    /// </summary>
    /// <param name="url">地址</param>
    /// <param name="buttonType">按钮类型</param>
    /// <param name="isPage">是否是页面</param>
    public PermissionAttribute(string url = default(string), ButtonType buttonType = default(ButtonType), bool isPage = true)
        base(typeof(RequiresPermissionAttributeExecutor))
    {
        Arguments = new object[] { new PermissionAuthorizationRequirement(url, buttonType, isPage) };
    }
    /// <summary>
    /// 构造器
    /// </summary>
    /// <param name="url">地址</param>
    /// <param name="buttonType">按钮类型</param>
    /// <param name="isPage">是否是页面</param>
    public PermissionAttribute(string url, byte buttonType, bool isPage = true)
        base(typeof(RequiresPermissionAttributeExecutor));
    {
        Arguments = new object[] { new PermissionAuthorizationRequirement(url, buttonType, isPage) };
    }
}

private class RequiresPermissionAttributeExecutor : Attribute, IAsyncResource
{
    private IPermissionStorageContainer _permissionStorage;
    private PermissionAuthorizationRequirement _requiredPermissions;
    public RequiresPermissionAttributeExecutor(IPermissionStorageContainer permissionStorage, PermissionAuthorizationRequirement requiredPermissions)
    {
        _permissionStorage = permissionStorage;
        _requiredPermissions = requiredPermissions;
    }
}
```

```
public async Task OnResourceExecutionAsync(ResourceExecutingContext context)
{
    string menuUrl = _requiredPermissions.UrlAndButtonType.Url;
    //判断用户权限
    if (string.IsNullOrEmpty(menuUrl))
    {
        //区域判断
        string area = context.RouteData.Values["area"].ToString();
        if (string.IsNullOrEmpty(area))
        {
            menuUrl = "/" + context.RouteData.Values["controller"] + "/";
        }
        else
        {
            menuUrl = "/" + area + "/" + context.RouteData.Values["controller"];
        }
    }
    menuUrl = menuUrl.Trim().ToLower();
    var dbpermission = await _permissionStorage.GetPermissionAsync();
    var menu = dbpermission.Menus.FirstOrDefault(m => m.MenuUrl != null);
    if (menu != null)//地址存在
    {
        if (_requiredPermissions.UrlAndButtonType.ButtonType == default(ButtonType))
        {
            await next();
        }
        else
        {
            byte buttonType = (byte)_requiredPermissions.UrlAndButtonType;
            if (menu.MenuButton.Select(m => m.ButtonType).Contains(buttonType))
            {
                await next();
            }
            else
            {
                //没有操作权限
                if (_requiredPermissions.UrlAndButtonType.IsPage)
                {
                    context.Result = new RedirectResult("/error/noauth");
                }
                else
                {
                    context.Result = new ContentResult()
                    {
                        Content = PermissionStatusCodes.Status2Unauthorized
                    };
                }
            }
        }
    }
}
```

```

        };
    }
    await context.Result.ExecuteResultAsync(context);
}
}
else
{
    //没有操作权限
    if (_requiredPermissions.UrlAndButtonType.IsPage)
    {
        context.Result = new RedirectResult("/error/noauth");
    }
    else
    {
        context.Result = new ContentResult()
        {
            Content = PermissionStatusCodes.Status2Unauthorized.ToString();
        };
    }
    await context.Result.ExecuteResultAsync(context);
}
}
}
}

```

在对于的页面添加过滤器即可，如下：

```

[HttpGet]
[Permission]
public async Task<IActionResult> Index(int pageIndex=1,int pageSize=10)
{
    var res = await _messageService.GetPageAsync(pageIndex, pageSize);
    return View(res);
}
[HttpGet]
[Permission("/Sys/Message/Index", ButtonType.View)]
public IActionResult Show()
{
    return View();
}

```

系统界面展示图：后台模板是之前从网上找的并自己简单改了一下，将就能看吧，实在不想花功夫在前端上面了@-^-@

This screenshot shows the 'User Management' section of a web application. The left sidebar includes 'User Management', 'Role Management', 'Department Management', 'System Management', 'Log Management', 'Microservice Management', 'Administrative Office', and 'Workshop'. The main content area has a search bar and a table listing users:

用户名	用户名	工号	创建时间	分配角色	分配部门	账号状态	操作
wms	wms	20180101	1970-01-01 12:00:12	管理员	系统	正常	查看
wangjiali	王五123	20180102	2017-05-27 01:48:42	管理员	系统	正常	查看
zhangsan	张三	20180103	2017-07-11 05:21:50	管理员	系统	正常	查看
lisi	李四aa	20180104	2017-07-11 05:22:03	管理员	系统	正常	查看
123	123	20180105	2017-07-11 05:22:14	管理员	系统	正常	查看

This screenshot shows the 'Role Management' section of the same web application. The left sidebar is identical to the previous screenshot. The main content area has a search bar and a table listing roles:

角色名称	备注	创建时间	分配资源	分配用户	状态	操作
系统管理员	系统管理员	2017-06-19 04:20:00			正常	查看

部门名称	部门编码	备注	操作
MS软件	001	MS软件	
总经办	002	总经理	
研发部	003	研发部	
销售部	004	销售部	

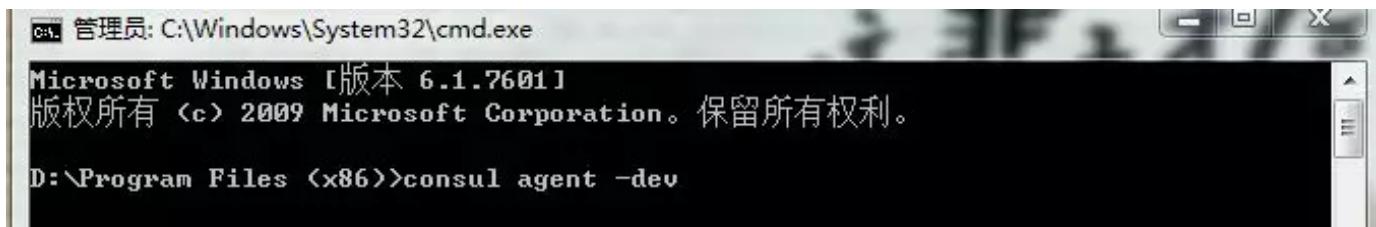
系统名称	系统备注	系统排序	系统编码	创建时间	系统状态	操作
行政办公系统	行政系统	1	be1c63a0-63aa-11e7-a221-f97d872f561b	2017-06-26 12:47:10		
微信系统	微信系统	2	d69fb3df-d342-41c9-ad9a-3faedbb5b0dc	2017-07-25 04:09:07		
工作流	工作流	3	c7526ad9-63aa-11e7-a221-f97d872f561b	2017-06-26 12:47:10		
财务系统	财务系统	4	c3cf9538-63aa-11e7-a221-f97d872f561b	2017-06-26 12:47:10		

菜单名称	URL	备注	操作权限	排序	状态
系统管理	/	123		1	正常
用户管理	/Sys/User/Index			2	正常
角色管理	/Sys/Role/Index	123		3	正常
角色编辑页	/Sys/Role/Show			4	正常
部门管理	/Sys/Dept/Index			5	正常
子系统管理	/Sys/System/Index	123		6	正常
菜单管理	/Sys/Resource/Index			7	正常
调度中心	/Schedule/Home/Index			8	正常
日志列表	/Sys/Log/Index			9	正常

运行步骤：

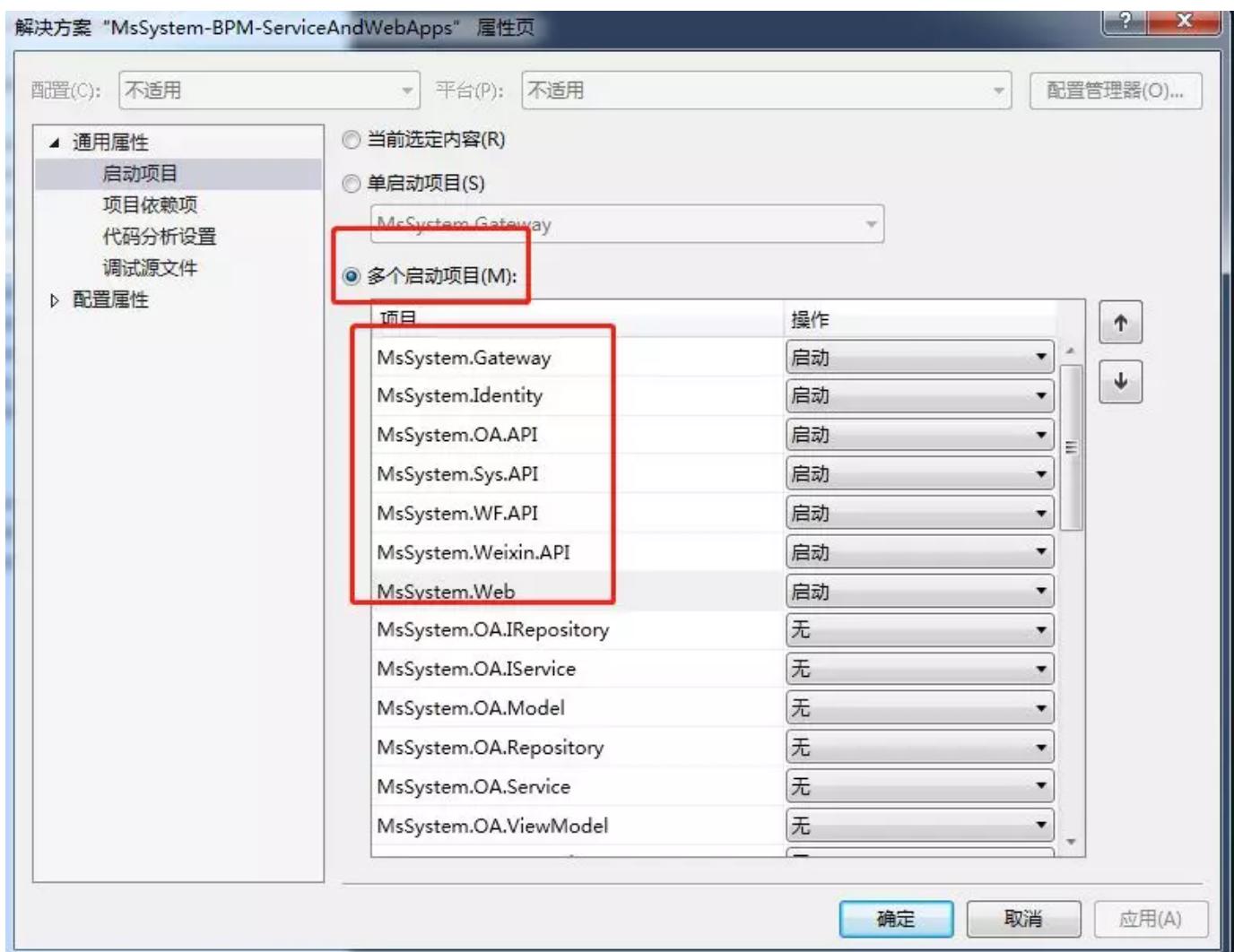
1、确保数据库mssystem和mssystemlog存在 github文档中

2、consul服务启动，如下回车运行



```
D:\Program Files (<x86>)>consul agent -dev
```

3、VS项目启动



管理员登录账号wms，密码：所有账号密码都是123

代码地址：

<https://github.com/wangmaosheng/MsSystem-BPM-ServiceAndWebApps>

如果觉得有点作用的话，可以 start 下，后续会持续更新

推荐阅读 (点击标题可跳转阅读)

[站点部署 IIS配置优化指南](#)

[iNeuOS云操作系统 .NET Core全系打造](#)

[ASP.NET Core 程序发布到 Centos](#)

看完本文有收获？请转发分享给更多人

关注「DotNet」加星标，提升.Net技能

DotNet

分享 .NET 相关技术干货 · 资讯 · 高薪职位 · 教程



微信号：iDotNet



长按识别二维码关注

伯乐在线旗下微信公众号

商务合作QQ：2302462408

好文章，我在看♡

[阅读原文](#)

