

乐翼教育云平台主要代码及简要说明

目录

单点登录	3
1. 登录接口	3
2. 接口调用日志	4
3. 接口配置	4
4. 路由配置	5
5. 邮件发送	6
6. 控制面板	7
7. 用户注册	8
8. 用户登录	12
9. 用户退出登录	16
乐翼教育云平台	18
1. 基类方法	18
2. 扩展方法	20
3. 数据表格显示	20
4. 全局消息通知	21
5. 菜单项加载	21
6. 权限管理	23
7. 组织管理	26
8. 课程管理	30
9. 人员管理	33
10. 云资源平台基础数据相关	35

11.	数据上传导入	41
12.	查询审计	41
13.	个人设定	44
乐翼云资源平台		46
1.	基础方法	46
2.	资源互动	53
3.	资源板块	54
4.	资源发布	58

单点登录

1. 登录接口

//单点登录接口 Key

```
private static readonly Guid SignApi = Guid.Parse("8646F44F-A854-4AF5-820F-BCC00E31BB52");
```

//根据第三方平台 Key 调用接口获取已登录用户信息

```
[HttpGet]
```

```
[Route("Sign/{key}")]
```

```
public dynamic Get(Guid key, Guid token)
```

```
{
```

```
    if (HomoryContext.Value.Api.Count(o => o.Id == SignApi && o.ProviderKey == key  
&& o.State < State.审核) > 0)
```

```
    {
```

```
        var providerId = HomoryContext.Value.Api.First(o => o.Id == TeacherApi &&  
o.ProviderKey == key && o.State < State.审核).ProviderId;
```

//记录接口调用日志

```
LogApi(TeacherApi, providerId);
```

```
if (HomoryContext.Value.UserOnline.Count(o => o.Id == token) == 0)
```

```
    return null;
```

```
var u = HomoryContext.Value.UserOnline.First(o => o.Id == token).User;
```

//返回接口数据

```
return new
```

```
{
```

```
    账号 = u.Account,
```

```
    姓名 = u.RealName,
```

```

        昵称 = u.DisplayName

    };

}

else

    return null;

}

<!DOCTYPE html>

<html>

<head runat="server">

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=Edge,Chrome=1" />

    <meta name="viewport" content="width=device-width,initial-scale=1,minimum-
scale=1,maximum-scale=1" />

    <title>北塘教育</title>

    <link href="../../Content/Semantic/css/semantic.min.css" rel="stylesheet" />

    <link href="../../Content/Homory/css/common.css" rel="stylesheet" />

    <link href="../../Content/Sso/css/sign.css" rel="stylesheet" />

    <script src="../../Content/jQuery/jquery.min.js"></script>

    <script src="../../Content/Semantic/javascript/semantic.min.js"></script>

    <script src="../../Content/Homory/js/common.js"></script>

    <script src="../../Content/Homory/js/notify.min.js"></script>

    <script src="../../Content/Sso/js/signOn.js"></script>

    <!--[if lt IE 9]>

```

```

        <script src="../../Content/Homory/js/respond.min.js"> </script>

    <![endif]-->

</head>

<body>

    <form id="form" runat="server">

        <telerik:RadScriptManager runat="server"> </telerik:RadScriptManager>

        <div class="ui two column page grid">

            <div class="ten wide column">

                

            </div>

            <div class="six wide column">

                

            </div>

        </div>

        <div class="ui two column stackable page grid">

            <div class="five wide column">

                <div class="ui form">

                    <div class="ui white segment">

                        <telerik:RadAjaxPanel ID="areaAction"
runat="server">

                            <div class="ui black inverted center
aligned segment">

                                <label>用户登录</label>

                            </div>

```

```

width80">账号</label>

<label class="ui blue ribbon label

<div class="field padTop15">

    <div class="ui left labeled icon

input">

    <input id="userName"

runat="server" type="text" value="" maxlength="256" />

    <i class="user icon"> </i>

    </div>

</div>

<label class="ui blue ribbon label

width80">密码</label>

<div class="field padTop15">

    <div class="ui left labeled icon

input">

    <input id="userPassword"

runat="server" type="password" maxlength="64" value="" />

    <i class="lock icon"> </i>

    </div>

</div>

<div class="ui right aligned column

padTop15">

    <div id="buttonAuto" class="ui

toggle checkbox">

    <input

name="autoPassword" type="checkbox">

```

```

<label
id="autoPasswordLabel">不记住密码</label>

</div>

</div>

<div class="ui divider"></div>

<div class="ui center aligned grid">

    <div class="column">

        <asp:Button
ID="buttonSign" runat="server" OnClientClick="doRemember();"
OnClick="buttonSign_OnClick" CssClass="circular ui blue button" Style="margin-left: 10px;
margin-right: 10px;" Text="登录"></asp:Button>

        <asp:Button
ID="buttonRegister" runat="server" OnClick="buttonRegister_OnClick" CssClass="circular ui
blue button" Style="margin-left: 10px; margin-right: 10px;" Text="注册"></asp:Button>

    </div>

</div>

</telerik:RadAjaxPanel>

</div>

</div>

<div class="eleven wide column">

    <div class="ui white segment">

        <telerik:RadImageGallery ID="gallery" runat="server"
ShowLoadingPanel="False" AllowPaging="false" Height="415px"
ImagesFolderPath="~/Common/配置/SsoSplash" DisplayAreaMode="Image"
LoopItems="true">

```

```

        <ThumbnailsAreaSettings Mode="ImageSlider"
ShowScrollbar="false" ShowScrollButtons="false" />

        <ImageAreaSettings NavigationMode="Zone"
NextImageButtonText="下一图" PrevImageButtonText="上一图" ShowDescriptionBox="false"
ShowNextPrevImageButtons="false" />

        <ClientSettings>

            <ClientEvents

OnImageGalleryCreated="galleryLoaded" />

            <AnimationSettings

SlideshowSlideDuration="6000">

                <NextImagesAnimation

Type="Random" Easing="Random" Speed="3000" />

                <PrevImagesAnimation

Type="Random" Easing="Random" Speed="3000" />

            </AnimationSettings>

        </ClientSettings>

        <ToolbarSettings ShowFullScreenButton="true"

ShowSlideshowButton="true" EnterFullScreenButtonText="全屏"
ExitFullScreenButtonText="退出全屏" PlayButtonText="播放" PauseButtonText="暂停"
ShowItemsCounter="false" Position="None" />

    </telerik:RadImageGallery>

</div>

</div>

</div>

<div class="ui center aligned page grid">

    <telerik:RadAjaxPanel ID="areaFavourite" runat="server">

        <div class="twelve wide column">

```



```

        <div id="signThumb" class="ui horizontal icon divider">

            <i id="signThumbButton" class="circular blue
inverted thumbs up icon signOnPointer"> </i>

            <div id="signThumbCount" class="floating ui
blue label width100">

                <asp:Label ID="FavouriteCount"
runat="server"> </asp:Label> <asp:Button ID="signThumbPost" runat="server"
CssClass="signHidden" OnClick="signThumbPost_OnClick" />

            </div>

        </div>

    </div>

</telerik:RadAjaxPanel>

</div>

<div class="ui four middle aligned column stackable page grid">

    <div class="four wide column">

        <div class="ui blue segment">

            <div class="ui inverted segment">

                <div class="ui blue header">

                    <i class="green cloud icon"> </i>

                    <div class="content signBreak"

style="font-size:16px;">

                        <asp:Label ID="ApplicationCount"
runat="server"> </asp:Label> 大应用模块

                    </div>

                </div>

            </div>

        </div>

    </div>

    <div class="sub header">平台一体化</div>

</div>

```

```

        </div>
    </div>
</div>
</div>
<div class="four wide column">
    <div class="ui green segment">
        <div class="ui inverted segment">
            <div class="ui green header">
                <i class="blue video icon"></i>
                <div class="content signBreak"
style="font-size:16px;">
                    <asp:Label ID="ResourceCount"
runat="server"></asp:Label>教育资源
            </div>
            <div class="sub header">资源多样
化</div>
        </div>
    </div>
</div>
</div>
</div>
</div>
<div class="four wide column">
    <div class="ui blue segment">
        <div class="ui inverted segment">
            <div class="ui blue header">

```

```

<i class="green users icon"></i>

<div class="content signBreak"

style="font-size:16px;">

<asp:Label ID="UserCount"

runat="server"></asp:Label>位教育用户

<div class="sub header">用户专业化</div>

</div>

</div>

</div>

</div>

</div>

<div class="four wide column">

<div class="ui green segment">

<div class="ui inverted segment">

<div class="ui green header">

<i class="blue dashboard icon"></i>

<div class="content signBreak"

style="font-size:16px;">

<asp:Label ID="VisitCount"

runat="server"></asp:Label>次统一登录

<div class="sub header">登录统一化</div>

</div>

</div>

```

```

        </div>

    </div>

</div>

<div class="ui center aligned page grid">

    <div class="center aligned column">

        

    </div>

</div>

</form>

</body>

</html>

```

2. 接口调用日志

```
protected void LogApi(Guid apiId, string providerId)
```

```

{
    //根据第三方传入的 Key 记录接口调用日志

    HomoryContext.Value.ApiLog.Add(new ApiLog
    {
        Id = HomoryContext.Value.GetId(),
        ApiId = apiId,
        ProviderId = providerId,
        Time = DateTime.Now
    });
}

```

```
        HomoryContext.Value.SaveChanges();  
    }  
}
```

3. 接口配置

```
public static class ApiConfig
```

```
{  
  
    public static void Register(HttpConfiguration config)  
    {  
  
        config.MapHttpAttributeRoutes();  
  
        //配置接口返回 Json 数据  
        config.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling =  
            ReferenceLoopHandling.Ignore;  
  
        config.Formatters.JsonFormatter.SerializerSettings.MissingMemberHandling =  
            MissingMemberHandling.Ignore;  
  
        config.Formatters.JsonFormatter.SerializerSettings.PreserveReferencesHandling  
            = PreserveReferencesHandling.Objects;  
  
        config.Formatters.Remove(config.Formatters.XmlFormatter);  
  
        //配置接口可跨域访问  
        config.EnableCors(new EnableCorsAttribute("*", "*", "*"));  
    }  
}
```

4. 路由配置

```
public static class RouteConfig
```

```
{  
  
    public static void RegisterRoutes(RouteCollection routes)  
    {  
  
        //映射路由
```

```

        var settings = new FriendlyUrlSettings { AutoRedirectMode =
RedirectMode.Permanent };

        routes.EnableFriendlyUrls(settings);

        routes.Ignore("{resource}.axd/{*pathInfo}");

    }
}

```

5. 邮件发送

```

public class Sntp
{
    public static bool SendEmail(Entities db, string title, string alias, string content, string
address, string sender)
    {
        try
        {
            //根据数据库设置初始化邮件发送参数

            var host = db.ApplicationPolicy.Where(o => o.Name == "SntpHost"
&& o.ApplicationId == Guid.Empty).FutureFirstOrDefault();

            var port = db.ApplicationPolicy.Where(o => o.Name == "SntpPort"
&& o.ApplicationId == Guid.Empty).FutureFirstOrDefault();

            var account = db.ApplicationPolicy.Where(o => o.Name ==
"SntpAccount" && o.ApplicationId == Guid.Empty).FutureFirstOrDefault();

            var password = db.ApplicationPolicy.Where(o => o.Name ==
"SntpPassword" && o.ApplicationId == Guid.Empty).FutureFirstOrDefault();

            var smtp = new SntpClient
            {
                EnableSsl = false,

```

```

        Host = host.Value.Value,
        Port = int.Parse(port.Value.Value),
//构建用户凭证
        Credentials = new NetworkCredential(account.Value.Value,
password.Value.Value)
    };
    var mm = new MailMessage
    {
        From = new MailAddress(account.Value.Value, sender,
Encoding.UTF8)
    };
    mm.To.Add(new MailAddress(address, alias, Encoding.UTF8));
    mm.Subject = title;
    mm.SubjectEncoding = Encoding.UTF8;
    mm.IsBodyHtml = true;
    mm.BodyEncoding = Encoding.UTF8;
    mm.Body = content;
//发送邮件
    smtp.Send(mm);
    return true;
}
catch
{
    return false;
}

```

```

    }
}
}

```

6. 控制面板

//加载应用平台

```

private void LoadSite(UserType userType)
{
    if (userType == UserType.内置)
    {
        var sitesX = HomoryContext.Value.Application.Where(o => o.Type ==
ApplicationType.平台 && o.State < State.审核);

        items.DataSource = sitesX.OrderBy(o => o.Ordinal).ToList();

        items.DataBind();

        return;
    }

    var sites = HomoryContext.Value.ApplicationRole.Where(o => o.UserType ==
userType).ToList().Select(o => o.Application).ToList()

    .Where(o => o.Type == ApplicationType.平台 && o.State < State.审核);

    items.DataSource = sites.OrderBy(o => o.Ordinal).ToList();

    items.DataBind();
}

<!DOCTYPE html>

<html>

<head runat="server">

```



```

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=Edge,Chrome=1" />

<meta name="viewport" content="width=device-width,initial-scale=1,minimum-
scale=1,maximum-scale=1" />

<title>面板</title>

<link href="../../Content/Semantic/css/semantic.min.css" rel="stylesheet" />

<link href="../../Content/Homory/css/common.css" rel="stylesheet" />

<link href="../../Content/Sso/css/sign.css" rel="stylesheet" />

<link href="../../Content/Sso/css/board.css" rel="stylesheet" />

<script src="../../Content/jQuery/jquery.min.js"></script>

<script src="../../Content/Semantic/javascript/semantic.min.js"></script>

<script src="../../Content/Homory/js/common.js"></script>

<script src="../../Content/Homory/js/notify.min.js"></script>

<script src="../../Content/Sso/js/board.js"></script>

<!--[if lt IE 9]>

    <script src="../../Content/Homory/js/respond.min.js"></script>

<![endif]-->

</head>

<body>

    <form id="form" runat="server">

        <telerik:RadScriptManager runat="server"></telerik:RadScriptManager>

        <div class="ui stackable page grid">

            <div class="column">

                <div id="bar" class="ui teal inverted center aligned segment">

```

```

        <asp:Image ID="icon" runat="server" CssClass="ui image" Width="60"
Height="60" />

        <label id="headInfo" runat="server">控制面板</label>

        <a href=" ../Go/SignOff" id="quit" class="ui animated tiny purple button floated
right">

            <div class="visible content">退出</div>

            <div class="hidden content">

                <i class="right sign out icon"></i>&nbsp;

            </div>

        </a>

    </div>

    </div>

    <div class="ui stackable center aligned page grid">

        <asp:Repeater ID="items" runat="server">

            <ItemTemplate>

                <div class="five wide column">

                    <div class="item signLink" data-url='<%# (string.Format("{0}?OnlineId={1}",
Eval("Home"), Session[Homory.Model.HomoryConstant.SessionOnlineId])) %>'>

                        <p class="ui image square200" style="background: none; text-align:
center;">

                            <img style="margin: auto;" onerror="this.src = '../Common/默认/应用.
png';" src='<%# Eval("Icon") %>' alt="" />

                        </p>

                        <p class="ui red large circular button">

```

```

        <%# Eval("Name") %>

    </p>

</div>

</div>

</ItemTemplate>

</asp:Repeater>

</div>

<div class="ui right aligned page grid">

    <div class="right aligned column">

        

    </div>

</div>

<style>

    html .image.ui {

        background: none;

    }

</style>

</form>

</body>

</html>

```

7. 用户注册

```

public dynamic UserRegister(Entities db, Guid id, string email, string password1, string
password2, string name,

```

```

//生成注册用户信息

```

```

string realName, string iconUrl, string description)
{
    dynamic output = new ExpandoObject();

    output.Ok = false;

    output.Data = new ExpandoObject();

    try
    {
        var open =
//读取用户注册相关策略
        db.ApplicationPolicy.Where(o => o.Name == "UserRegistration" &&
o.ApplicationId == Guid.Empty)

            .FutureFirstOrDefault();

        var mRegex =

            db.ApplicationPolicy.Where(o => o.Name == "UserEmailRegex" &&
o.ApplicationId == Guid.Empty).FutureFirstOrDefault();

        var length =

            db.ApplicationPolicy.Where(o => o.Name == "UserPasswordLength" &&
o.ApplicationId == Guid.Empty)

                .FutureFirstOrDefault();

        var count = db.User.Where(o => o.Account == email && o.State < State.审核)
.FutureCount();

        if (!bool.Parse(open.Value.Value))
        {
            output.Data.Message = "用户注册功能暂未开放";

            output.Data.Parameter = string.Empty;

```

```

        return output;
    }

    if (string.IsNullOrEmpty(email) || !new
Regex(mRegex.Value.Value).IsMatch(email))

    {
        output.Data.Message = "请输入.com/.cn 结尾的电子邮箱";
        output.Data.Parameter = "email";
        return output;
    }

//判断用户提交的注册信息的合法性

    if (string.IsNullOrEmpty(password1) || string.IsNullOrEmpty(password2) ||
        !password1.Equals(password2, StringComparison.Ordinal) || password1.Length <
int.Parse(length.Value.Value))

    {
        output.Data.Message = string.Format("请输入不少于{0}位的密码，并确保两次输入
的密码一致", length);
        output.Data.Parameter = "password";
        return output;
    }

    string key, salt;

//加密密码

    var password = HomoryCryptor.Encrypt(password1, out key, out salt);

    if (count.Value == 0)

    {
        var user = new User

```

```

{
    Id = id,
    Account = email,
    RealName = string.IsNullOrEmpty(realName) ?
(string.IsNullOrEmpty(name) ? email : name) : realName,
    DisplayName = string.IsNullOrEmpty(name) ? email : name,
    Icon = iconUrl,
    Stamp = Guid.NewGuid(),
    Password = password,
    PasswordEx = null,
    CryptoKey = key,
    CryptoSalt = salt,
    Type = UserType.注册,
    State = State.默认,
    Ordinal = 0,
    Description = string.IsNullOrEmpty(description) ? null : description
};

//保存用户信息并返回用户数据
db.User.Add(user);
db.SaveChanges();
output.Data.Entity = user;
output.Ok = true;
return output;
}

```

```

        output.Data.Message = "电子邮件已被注册";

        output.Data.Parameter = string.Empty;

        return output;
    }

    catch (Exception exception)
    {
        output.Data.Message = exception.Message;

        return output;
    }
}

<!DOCTYPE html>

<html>

<head runat="server">

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=Edge,Chrome=1" />

    <meta name="viewport" content="width=device-width,initial-scale=1,minimum-
scale=1,maximum-scale=1" />

    <title>注册</title>

    <link href="../../Content/Semantic/css/semantic.min.css" rel="stylesheet" />

    <link href="../../Content/Homory/css/common.css" rel="stylesheet" />

    <link href="../../Content/Sso/css/sign.css" rel="stylesheet" />

    <link href="../../Content/Sso/css/register.css" rel="stylesheet" />

    <script src="../../Content/jQuery/jquery.min.js"></script>

```

```

<script src="../../Content/Semantic/javascript/semantic.min.js"></script>

<script src="../../Content/Homory/js/common.js"></script>

<script src="../../Content/Homory/js/notify.min.js"></script>

<!--[if lt IE 9]>

    <script src="../../Content/Homory/js/respond.min.js"></script>

<![endif]-->

</head>

<body>

    <form id="form" runat="server">

        <telerik:RadScriptManager runat="server"></telerik:RadScriptManager>

        <telerik:RadAjaxManager ID="ajaxManager" runat="server"
EnablePageHeadUpdate="false">

            <AjaxSettings>

                <telerik:AjaxSetting AjaxControlID="ajaxManager">

                    <UpdatedControls>

                        <telerik:AjaxUpdatedControl ControlID="viewer"

/>

                        <telerik:AjaxUpdatedControl ControlID="upload"

/>

                    </UpdatedControls>

                </telerik:AjaxSetting>

            </AjaxSettings>

        </telerik:RadAjaxManager>

        <div class="ui page grid">

            <div class="column">

```



```

        
    </div>
</div>
<div class="ui page grid">
    <div class="column">
        <div class="ui teal inverted center aligned segment">
            <label>用户注册</label>
        </div>
    </div>
</div>
<div class="ui page grid">
    <div class="column">
        <div class="ui center aligned column">
            <div class="ui small steps">
                <div class="ui active step">
                    填写注册信息
                </div>
                <div class="ui step">
                    验证注册邮箱
                </div>
                <div class="ui step">
                    用户注册成功
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>

    </div>

</div>

<telerik:RadAjaxPanel ID="areaAction" runat="server">

    <div class="ui two column stackable page grid">

        <div class="column">

            <div class="ui form segment">

                <label class="ui teal ribbon label width100">电子
邮箱</label>

                <div class="field padTop15">

                    <div class="ui left labeled icon input">

                        <input id="userEmail"
runat="server" type="text" value="" maxlength="256" />

                        <i class="mail icon"></i>

                        <div class="ui corner label">

                            <i class="icon asterisk"></i>

                        </div>

                    </div>

                </div>

            </div>

            <div class="ui teal ribbon label width100">密码
</label>

            <div class="field padTop15">

                <div class="ui left labeled icon input">

                    <input id="userPassword"
type="password" runat="server" value="" maxlength="256" />

```

```

        <i class="lock icon"> </i>

        <div class="ui corner label">

            <i class="icon asterisk"> </i>

        </div>

    </div>

</div>

<label class="ui teal ribbon label width100">密码
确认</label>

<div class="field padTop15">

    <div class="ui left labeled icon input">

        <input id="userPasswordRepeat"
type="password" value="" runat="server" maxlength="256" />

        <i class="lock icon"> </i>

        <div class="ui corner label">

            <i class="icon asterisk"> </i>

        </div>

    </div>

</div>

<label class="ui teal ribbon label width100">昵称

</label>

<div class="field padTop15">

    <div class="ui left labeled icon input">

        <input id="userName" type="text"
value="" maxlength="64" runat="server" />

        <i class="user icon"> </i>

```

```

        </div>
    </div>
</div>
</div>
<div class="column">
    <div class="ui form segment">
        <label class="ui teal ribbon label width100">真实
姓名</label>

        <div class="field padTop15">
            <div class="ui left labeled icon input">
                <input id="userRealName"
type="text" value="" maxlength="16" runat="server" />
                <i class="phone icon"></i>
            </div>
        </div>
        <label class="ui teal ribbon label width100">个人
头像</label>

        <div class="field padTop15">
            <div>
                <telerik:RadScriptBlock
runat="server">

                <script
type="text/javascript">

                function
OnClientFilesUploaded(sender, args) {

```

```

$find('<%=ajaxManager.ClientID %>').ajaxRequest();

        }

    </script>

</telerik:RadScriptBlock>

<asp:Image ID="viewer"

runat="server" Width="100" Height="100" />

        <telerik:RadAsyncUpload
ID="upload" runat="server" MultipleFileSelection="Disabled" Skin="BlackMetroTouch"
HideFileInput="true" AutoAddFileInputs="false" TemporaryFolder="~/Common/头像/用户"
TargetFolder="~/Common/头像/用户" Localization-Cancel="取消" Localization-Remove="移除"
Localization-Select="选择" OnClientFilesUploaded="OnClientFilesUploaded"
OnFileUploaded="upload_FileUploaded">

        </telerik:RadAsyncUpload>

    </div>

</div>

<label class="ui teal ribbon label width100">个人
描述</label>

    <div class="field padTop15">

        <div class="ui left labeled icon input">

            <input id="userDescription"

runat="server" type="text" value="" maxlength="1024" />

            <i class="pencil icon"></i>

        </div>

    </div>

</div>

</div>

```



```

        var user = db.User.SingleOrDefault(o => o.Account == account && o.State <
State.删除);

        if (user == null ||

            !password.Equals(HomoryCryptor.Decrypt(user.Password, user.CryptoKey,
user.CryptoSalt),

                StringComparison.OrdinalIgnoreCase))

        {

            output.Data.Message = "请输入正确的账号和密码";

            return output;

        }

        if (user.State == State.停用)

        {

            output.Data.Message = "用户已被停用";

            return output;

        }

        if (user.State == State.默认 || user.State == State.审核)

        {

            Session[HomoryConstant.SessionRegisterId] = user.Id;

            output.Ok = true;

            output.Data.Redirect = true;

            output.Data.RedirectUrl = "SsoVerify".FromHomoryConfig();

            return output;

        }

        //生成用户在线信息

```

```
var online = user.UserOnline.SingleOrDefault();
```

```
if (online == null)
```

```
{
```

```
    online = new UserOnline
```

```
    {
```

```
        Id = db.GetId(),
```

```
        UserId = user.Id,
```

```
        TimeStamp = DateTime.Now
```

```
    };
```

```
    db.UserOnline.Add(online);
```

```
}
```

```
else
```

```
{
```

```
    online.TimeStamp = DateTime.Now;
```

```
}
```

```
db.SaveChanges();
```

```
//设置用户凭证
```

```
var cookie = new HttpCookie(HomoryConstant.CookieOnlineId,  
online.Id.ToString().ToUpper());
```

```
var expire = int.Parse(db.ApplicationPolicy.Single(o => o.Name ==  
"UserCookieExpire" && o.ApplicationId == Guid.Empty).Value);
```

```
cookie.Expires = DateTime.Now.AddMinutes(expire);
```

```
Response.SetCookie(cookie);
```

```
Session[HomoryConstant.SessionOnlineId] = online.Id;
```



```

output.Ok = true;

output.Data.Redirect = true;

var query = Server.UrlDecode(Request.QueryString["SsoRedirect"]);

if (string.IsNullOrEmpty(Request.QueryString["SsoRedirect"]))
{
    output.Data.RedirectUrl = string.Format("SsoRedirect".FromHomoryConfig(),
user.Account, HomoryCryptor.Decrypt(user.Password, user.CryptoKey, user.CryptoSalt));
}
else
{
    string url;

    if (!string.IsNullOrEmpty(Request.QueryString["OnlineId"]))
    {
        url = query;
    }
    else
    {
        if (query.IndexOf('&') <= 0)
        {
            url = string.Format("{0}?OnlineId={1}", query, user.UserOnline.Single().Id);
        }
        else
        {
            var index = query.IndexOf('&');

```

```

        url =
            Server.UrlDecode(query.Remove(index, 1)
                .Insert(index, string.Format("?OnlineId={0}&",
user.UserOnline.Single().Id)));
    }
}

//根据 Url 信息跳转
        output.Data.RedirectUrl = string.IsNullOrEmpty(url) ?
string.Format("SsoRedirect".FromHomoryConfig(), user.Account,
HomoryCryptor.Decrypt(user.Password, user.CryptoKey, user.CryptoSalt)) : url;
    }

    return output;
}

catch (Exception exception)
{
    output.Data.Message = exception.Message;

    return output;
}
}

```

9. 用户退出登录

```

public bool UserSignOff(Entities db)
{
    try
    {
        //清空 Cookie 和 Session
    }
}

```

```

var oid = string.Empty;

    if (Session[HomoryConstant.SessionOnlineId] != null)
    {
        oid = Session[HomoryConstant.SessionOnlineId].ToString();
    }

    else if (Request.Cookies.AllKeys.Contains(HomoryConstant.CookieOnlineId))
    {
        var httpCookie = Request.Cookies[HomoryConstant.CookieOnlineId];

        if (httpCookie != null) oid = httpCookie.Value;
    }

    if (string.IsNullOrEmpty(oid))
    {
        return true;
    }

```

//注销数据库用户登录信息

```

var onlineGuid = Guid.Parse(oid);

db.UserOnline.Where(o => o.Id == onlineGuid).Delete();

db.SaveChanges();

Session.Remove(HomoryConstant.SessionOnlineId);

if (Request.Cookies.AllKeys.Contains(HomoryConstant.CookieOnlineId))
{
    var cookie = Request.Cookies[HomoryConstant.CookieOnlineId];

    if (cookie != null)
    {

```

```

        cookie.Expires = DateTime.Now.AddSeconds(-1);

        HttpContext.Current.Response.SetCookie(cookie);
    }

}

return true;

}

catch

{

return false;

}

}

}

```

乐翼教育云平台

1. 基类方法

//用户是否在线

```

protected bool IsOnline
{
    get { return Session[HomoryCoreConstant.SessionUserId] != null; }
}

```

//获取当前用户

```

protected User CurrentUser
{
    get
    {
        var id = (Guid)Session[HomoryCoreConstant.SessionUserId];
        return HomoryContext.Value.User.Single(o => o.Id == id);
    }
}

```

```
    }  
}
```

//当前用户的学校信息

protected Department CurrentCampus

```
{  
    get  
    {  
        if (_campus != null) return _campus;  
        var department = CurrentUser.DepartmentUser.SingleOrDefault(o => o.Type ==  
DepartmentUserType.部门主职教师 && o.State == State.启用);  
        if (department == null) return null;  
        _campus = department.TopDepartment;  
        return _campus;  
    }  
}
```

//获取当前用户的权限列表

protected List<string> CurrentRights

```
{  
    get  
    {  
        if (_rights != null) return _rights;  
        if (CurrentUser.State == State.内置)  
        {  
            _rights =  
                HomoryContext.Value.Right.Where(o => o.ApplicationId ==  
HomoryCoreConstant.ApplicationKey)  
                .Select(o => o.Name)  
                .ToList();  
        }  
    }  
}
```

```

    }

    else

    {

        var role = CurrentUser.UserRole;

        _rights = role.Count == 0

            ? new[] { HomoryCoreConstant.RightEveryone }.ToList()

            : role.Where(o => o.State < State.审核)

                .ToList()

                .Join(HomoryContext.Value.RoleRight.Where(o => o.State < State.审核), o =>

o.RoleId, o => o.RoleId,

                    (o1, o2) => o2.RightName)

                .ToList()

                .Union(new[] { HomoryCoreConstant.RightEveryone })

                .ToList();

    }

    return _rights;

}

}

```

2. 扩展方法

//记录用户操作日志

```

public static void Operated(this Entities db, OperationType operationType, User user,
    Department campus)

    {

        if (user == null || campus == null)

            return;
    }

```

```

var ol = new OperationLog();

ol.Id = db.GetId();

ol.Time = DateTime.Now;

ol.Type = operationType;

ol.UserId = user.Id;

ol.CampusId = campus.Id;

db.OperationLog.Add(ol);

db.SaveChanges();

}

```

3. 数据表格显示

//列表数据项目创建函数

```

protected virtual void grid_ItemCreated(object sender, GridItemEventArgs e)
{
    if (!(e.Item is GridDataItem) || e.Item.DataItem == null) return;

    var state = (e.Item.DataItem as dynamic).State;

    if (state == State.启用 || state == State.内置)
    {
        e.Item.ForeColor = HomoryCoreConstant.Color 启用;
    }

    else if (state == State.停用)
    {
        e.Item.ForeColor = HomoryCoreConstant.Color 停用;
    }

    else if (state == State.默认 || state == State.审核)

```

```

        {
            e.Item.ForeColor = HomoryCoreConstant.Color 审核;
        }
    }
}

<!DOCTYPE html>

<html>

<head runat="server">

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=Edge,Chrome=1" />

    <meta name="viewport" content="width=device-width,initial-scale=1,minimum-
scale=1,maximum-scale=1" />

    <title>基础平台</title>

    <link href="../../Content/Semantic/css/semantic.min.css" rel="stylesheet" />

    <link href="../../Content/Homory/css/common.css" rel="stylesheet" />

    <link href="../../Content/Core/css/common.css" rel="stylesheet" />

    <!--    <link href="../../Content/Core/css/complex.css" rel="stylesheet" />--%>

    <script src="../../Content/jQuery/jquery.min.js"></script>

    <script src="../../Content/Semantic/javascript/semantic.min.js"></script>

    <script src="../../Content/Homory/js/common.js"></script>

    <script src="../../Content/Homory/js/notify.min.js"></script>

    <!--[if lt IE 9]>

        <script src="../../Content/Homory/js/respond.min.js"></script>

    <![endif]-->

```



```

</head>

<body>

  <form id="formHome" runat="server">

    <div>

      <homory:SideBar runat="server" ID="SideBar" />

    </div>

    <telerik:RadAjaxLoadingPanel ID="loading" runat="server">

      <i class="ui huge teal loading icon" style="margin-top: 50px;"> </i>

      <div> &nbsp;</div>

      <div style="color: #564F8A; font-size: 16px;">正在加载 请稍候....</div>

    </telerik:RadAjaxLoadingPanel>

    <script>

      function mngRequestStarted(ajaxManager, eventArgs) {

        if (eventArgs.get_eventTarget().indexOf("ExportToExcel") >= 0)

          eventArgs.set_enableAjax(false);

      }

    </script>

    <telerik:RadAjaxPanel ID="panel" runat="server" CssClass="ui left aligned stackable
page grid" Style="margin: 0; padding: 0;" LoadingPanelID="loading" ClientEvents-
OnRequestStart="mngRequestStarted">

      <div class="five wide column">

        <telerik:RadComboBox ID="combo" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="combo_SelectedIndexChanged" DataTextField="Name"
DataValueField="Id" Label="选择学校 : " Width="220px" Filter="Contains"
MarkFirstMatch="true" AllowCustomText="true" Height="202px">

```

```

        <ItemTemplate>

            <span><%#
GenerateTreeName((Homory.Model.Department)Container.DataItem, Container.Index, 0)
%></span>

        </ItemTemplate>

    </telerik:RadComboBox>

</div>

<div class="six wide center aligned column">

    <telerik:RadSearchBox ID="peek" runat="server" OnSearch="peek_Search"
EmptyMessage="查找教师...." EnableAutoComplete="false">

        </telerik:RadSearchBox>

    </div>

<div class="five wide right aligned column">

    &nbsp;

</div>

<div class="sixteen wide column">

    <table>

        <tr class="coreTop">

            <td rowspan="2">

                <telerik:RadTreeView ID="tree" runat="server" CssClass="coreLeft"
DataTextField="Name" DataValueField="Id" DataFieldID="Id" DataFieldParentID="ParentId"
OnNodeClick="tree_NodeClick">

                    <NodeTemplate>

                        <span><%# ForceTreeName(Container.DataItem as
Homory.Model.Department) %></span>

                    </NodeTemplate>

```

```

        </telerik:RadTreeView>

    </td>

    <td>

        <telerik:RadListView ID="view" runat="server" ItemPlaceholderID="holder"
OnNeedDataSource="view_NeedDataSource">

            <LayoutTemplate>

                <asp:Panel ID="innerPanel" runat="server" CssClass="ui left middle
aligned stackable grid" Style="border: solid 1px #828282; padding: 20px; margin: 0px;"
Visible='<%# ActionTeachers.Count > 0 %>'>

                    <div class="sixteen wide column" style="margin: 0; padding: 0;">

                        <asp:PlaceHolder ID="holder" runat="server"></asp:PlaceHolder>

                    </div>

                    <div class="sixteen wide column" style="margin: 10px 0 0 0;
padding: 0;">

                        <asp:Button ID="btnM" runat="server" CssClass="ui teal mini
button" Text="主职设定" OnClick="btnM_Click"></asp:Button>

                        <asp:Button ID="btnP" runat="server" CssClass="ui teal mini
button" Text="兼职设定" OnClick="btnP_Click"></asp:Button>

                        <asp:Button ID="btnB" runat="server" CssClass="ui teal mini
button" Text="借调设定" OnClick="btnB_Click" OnPreRender="btnB_Load"></asp:Button>

                        <asp:Button ID="btnV" runat="server" CssClass="ui teal mini
button" Text="可访设定" OnClick="btnV_Click" OnPreRender="btnV_Load"></asp:Button>

                        <asp:Button ID="btnR" runat="server" CssClass="ui teal mini
button" Text="密码重置" OnClick="btnR_Click"></asp:Button>

                    </div>

                </asp:Panel>

            </LayoutTemplate>

```

```

        <ItemTemplate>

            <telerik:RadButton ID="actionButton" runat="server" Text='<%#
Eval("RealName") %>' Value='<%# Eval("Id") %>'
OnClick="actionButton_Click"></telerik:RadButton>

        </ItemTemplate>

    </telerik:RadListView>

</td>

</tr>

<tr class="coreTop">

    <td>

        <div id="MainPanel" runat="server">

            <h6 class="ui teal header left floated"><i class="ui teal circle
icon"></i>主职</h6>

            <div style="clear: both;"></div>

            <telerik:RadGrid ID="grid" runat="server" AllowPaging="true"
BackImageUrl="../../Images/Common/BG.gif" AutoGenerateColumns="false"
LocalizationPath="../../Language" AllowSorting="True" PageSize="25" GridLines="None"
OnNeedDataSource="grid_NeedDataSource"
OnBatchEditCommand="grid_BatchEditCommand" OnItemCreated="grid_ItemCreated">

                <MasterTableView EditMode="Batch" DataKeyNames="Id"
CommandItemDisplay="Top" HorizontalAlign="NotSet" ShowHeader="true"
ShowHeadersWhenNoRecords="true" NoMasterRecordsText="">

                    <BatchEditingSettings EditType="Row"
OpenEditingEvent="DbClick" />

                    <CommandItemSettings ShowExportToExcelButton="true"
ExportToExcelText="导出" />

                    <HeaderStyle HorizontalAlign="Center" />

```

```

<Columns>

    <telerik:GridTemplateColumn
UniqueName="ActionSelectionColumn" ItemStyle-HorizontalAlign="Center" HeaderStyle-
Width="48" ItemStyle-Width="48">

        <HeaderTemplate>

            <asp:CheckBox ID="actSelX" runat="server" Checked='<%#
ContainsAllActSel() %>' OnCheckedChanged="actSel_CheckedChangedX"
AutoPostBack="true" />

        </HeaderTemplate>

        <ItemTemplate>

            <asp:CheckBox ID="actSel" runat="server" ItemID='<%#
Eval("Id").ToString() %>' Checked='<%# ActionTeachers.Contains((Guid)Eval("Id")) %>'
OnCheckedChanged="actSel_CheckedChanged" AutoPostBack="true" />

        </ItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="48" ItemStyle-Width="48" HeaderText="序号
*" DataField="PriorOrdinal" SortExpression="PriorOrdinal" UniqueName="PriorOrdinal">

        <ItemTemplate>

            <asp:Label ID="Label1" runat="server" Text='<%#
Eval("PriorOrdinal").ToString().PadLeft(2).Replace(" ", "0") %>'> </asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadNumericTextBox ID="PriorOrdinal"
runat="server" Width="40" EnabledStyle-HorizontalAlign="Center" MinValue="1"
MaxValue="99" AllowOutOfRangeAutoCorrect="true" Value='<%# Bind("PriorOrdinal") %>'>

                <NumberFormat DecimalDigits="0"

                AllowRounding="true" />

```

```

        </telerik:RadNumericTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="登录
账号 *" DataField="Account" SortExpression="Account" UniqueName="Account">

    <ItemTemplate>

        <asp:Label ID="Label2" runat="server" Text='<%#
Eval("Account") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Account" runat="server"
Width="72" EnabledStyle-HorizontalAlign="Center" MaxLength="16" Text='<%#
Bind("Account") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="72" ItemStyle-Width="72" HeaderText="姓名
*" DataField="RealName" SortExpression="RealName" UniqueName="RealName">

    <ItemTemplate>

        <asp:Label ID="Label3" runat="server" Text='<%#
Eval("RealName") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

```

```

        <telerik:RadTextBox ID="RealName" runat="server"
Width="64" EnabledStyle-HorizontalAlign="Center" MaxLength="16" Text='<%#
Bind("RealName") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60" HeaderText="状态
" DataField="State" SortExpression="State" UniqueName="State">

    <ItemTemplate>

        <asp:Label ID="Label4" runat="server" Text='<%#
Eval("State") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadComboBox ID="State" runat="server"
Width="52" EnableTextSelection="true" Text='<%# Bind("State") %>'>

            <Items>

                <telerik:RadComboBoxItem Text="" Value="-1" />

                <telerik:RadComboBoxItem Text="启用" Value="1" />

                <telerik:RadComboBoxItem Text="停用" Value="4" />

                <telerik:RadComboBoxItem Text="删除" Value="5" />

            </Items>

        </telerik:RadComboBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

```

```

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60"
HeaderText="同步" DataField="Sync" SortExpression="Sync" UniqueName="Sync">

        <ItemTemplate>

            <asp:Label ID="Label7" runat="server" Text='<%#
(((bool)Eval("Sync")) ? "是" : "否") %>'></asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadComboBox ID="Sync" runat="server"
Width="52" EnableTextSelection="true" Text='<%# Bind("Sync") %>'>

                <Items>

                    <telerik:RadComboBoxItem Text="" Value="-1" />

                    <telerik:RadComboBoxItem Text="是" Value="True" />

                    <telerik:RadComboBoxItem Text="否" Value="False" />

                </Items>

            </telerik:RadComboBox>

        </EditItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="手机
号码 *" DataField="Phone" SortExpression="Phone" UniqueName="Phone">

        <ItemTemplate>

            <asp:Label ID="Label8" runat="server" Text='<%#
Eval("Phone") %>'></asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

```



```

        <telerik:RadTextBox ID="Phone" runat="server" Width="72"
EnabledStyle-HorizontalAlign="Center" MaxLength="11" Text='<%# Bind("Phone") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-HorizontalAlign="Left"
HeaderStyle-Width="100" ItemStyle-Width="100" HeaderText="电子邮件" DataField="Email"
SortExpression="Email" UniqueName="Email">

    <ItemTemplate>

        <asp:Label ID="Label9" runat="server" Text='<%#
Eval("Email") %>' ToolTip='<%# Eval("Email") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Email" runat="server" Width="92"
EnabledStyle-HorizontalAlign="Center" MaxLength="32" Text='<%# Bind("Email") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="128" ItemStyle-Width="128" HeaderText="身
份证号" DataField="IDCard" SortExpression="IDCard" UniqueName="IDCard">

    <ItemTemplate>

        <asp:Label ID="Label10" runat="server" Text='<%#
Eval("IDCard") %>' Font-Italic='<%# Eval("IDCard") != null &&
Eval("IDCard").ToString().Length != 18 %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

```

```

        <telerik:RadTextBox ID="IDCard" runat="server"
EnabledStyle-HorizontalAlign="Center" Width="120" MaxLength="18" Text='< %#
Bind("IDCard") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60" HeaderText="性别
" DataField="Gender" SortExpression="Gender" UniqueName="Gender">

    <ItemTemplate>

        <asp:Label ID="Label17" runat="server" Text='< %#
((bool?)Eval("Gender")).HasValue ? (((bool?)Eval("Gender")).Value ? "男" : "女") : string.Empty
%>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadComboBox ID="Gender" runat="server"
Width="52" EnableTextSelection="true" Text='< %# Bind("Gender") %>'>

            <Items>

                <telerik:RadComboBoxItem Text="" Value="-1" />

                <telerik:RadComboBoxItem Text="男" Value="True" />

                <telerik:RadComboBoxItem Text="女" Value="False" />

            </Items>

        </telerik:RadComboBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

```

```

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="88" ItemStyle-Width="88"
HeaderText="出生日期" DataField="Birthday" SortExpression="Birthday"
UniqueName="Birthday">

    <ItemTemplate>

        <asp:Label ID="Label18" runat="server" Text='<##
((DateTime?)Eval("Birthday")).HasValue ? ((DateTime)Eval("Birthday")).ToString("yyyy-MM-
dd") : string.Empty %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadDatePicker ID="Birthday" runat="server"
Width="80" DateInput-DateFormat="yyyy-MM-dd" LocalizationPath=" ../Language"
SelectedDate='<## Bind("Birthday") %>'> </telerik:RadDatePicker>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="48" ItemStyle-Width="48" HeaderText="民族
" DataField="Nationality" SortExpression="Nationality" UniqueName="Nationality">

    <ItemTemplate>

        <asp:Label ID="Label19" runat="server" Text='<##
Eval("Nationality") %>' ToolTip='<## Eval("Nationality") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Nationality" runat="server"
Width="40" EnabledStyle-HorizontalAlign="Center" MaxLength="16" Text='<##
Bind("Nationality") %>'>

    </telerik:RadTextBox>

</EditItemTemplate>

```

```

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="68" ItemStyle-Width="68" HeaderText="籍贯"
" DataField="Birthplace" SortExpression="Birthplace" UniqueName="Birthplace">

    <ItemTemplate>

        <asp:Label ID="Label20" runat="server" Text='< %#
Eval("Birthplace") %>' ToolTip='< %# Eval("Birthplace") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Birthplace" runat="server"
Width="60" EnabledStyle-HorizontalAlign="Center" MaxLength="16" Text='< %#
Bind("Birthplace") %>'>

    </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-HorizontalAlign="Left"
HeaderStyle-Width="128" ItemStyle-Width="128" HeaderText="现居住地"
DataField="Address" SortExpression="Address" UniqueName="Address">

    <ItemTemplate>

        <asp:Label ID="Label21" runat="server" Text='< %#
Eval("Address") %>' ToolTip='< %# Eval("Address") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Address" runat="server"
Width="120" EnabledStyle-HorizontalAlign="Center" MaxLength="128" Text='< %#
Bind("Address") %>'>

    </telerik:RadTextBox>

```

```

        </EditItemTemplate>

        </telerik:GridTemplateColumn>

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60" HeaderText="在编
" DataField="PerStaff" SortExpression="PerStaff" UniqueName="PerStaff">

            <ItemTemplate>

                <asp:Label ID="Label22" runat="server" Text='<%#
((bool?)Eval("PerStaff")).HasValue ? (((bool?)Eval("PerStaff")).Value ? "是" : "否") : string.Empty
%>'> </asp:Label>

            </ItemTemplate>

            <EditItemTemplate>

                <telerik:RadComboBox ID="PerStaff" runat="server"
Width="52" EnableTextSelection="true" Text='<%# Bind("PerStaff") %>'>

                    <Items>

                        <telerik:RadComboBoxItem Text="" Value="-1" />

                        <telerik:RadComboBoxItem Text="是" Value="True" />

                        <telerik:RadComboBoxItem Text="否" Value="False" />

                    </Items>

                </telerik:RadComboBox>

            </EditItemTemplate>

        </telerik:GridTemplateColumn>

    </Columns>

    <PagerStyle Mode="NextPrevAndNumeric"
PageSizes="10,20,50,100" Position="Bottom" PageSizeControlType="RadComboBox"
AlwaysVisible="true" PagerTextFormat="{4} 第{0}页，共{1}页；第{2}-{3}项，共{5}项" />

</MasterTableView>

```

```

        <ExportSettings Excel-Format="ExcelML" FileName="Teachers" Excel-
FileExtension="xls" IgnorePaging="true" ExportOnlyData="true"
OpenInNewWindow="true"> </ExportSettings>

    </telerik:RadGrid>

    <div class="ui divider"> </div>

    <h6 class="ui purple header left floated"> <i class="ui purple circle
icon"> </i> 兼职 </h6>

    <div style="clear: both;"> </div>

    <telerik:RadGrid ID="gridX" runat="server" AllowPaging="true"
AutoGenerateColumns="false" LocalizationPath="../Language" AllowSorting="True"
PageSize="10" GridLines="None" OnNeedDataSource="gridX_NeedDataSource"
OnBatchEditCommand="gridX_BatchEditCommand" OnItemCreated="grid_ItemCreated">

        <MasterTableView EditMode="Batch" DataKeyNames="Id"
CommandItemDisplay="Top" HorizontalAlign="NotSet" ShowHeader="true"
ShowHeadersWhenNoRecords="true" NoMasterRecordsText="">

            <BatchEditingSettings EditType="Row"
OpenEditingEvent="DbClick" />

            <CommandItemSettings
ShowAddNewRecordButton="False"> </CommandItemSettings>

            <HeaderStyle HorizontalAlign="Center" />

            <Columns>

                <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="48" ItemStyle-Width="48" HeaderText="序号
*" DataField="MinorOrdinal" SortExpression="MinorOrdinal" UniqueName="MinorOrdinal">

                    <ItemTemplate>

                        <asp:Label ID="Label12" runat="server" Text='<%#
Eval("MinorOrdinal").ToString().PadLeft(2).Replace(" ", "0") %>'> </asp:Label>

                    </ItemTemplate>

```

```

        <EditItemTemplate>

            <telerik:RadNumericTextBox ID="MinorOrdinal"
runat="server" EnabledStyle-HorizontalAlign="Center" Width="40" MinValue="1"
MaxValue="99" AllowOutOfRangeAutoCorrect="true" Value='<%# Bind("MinorOrdinal")
%>'>

                <NumberFormat DecimalDigits="0"

AllowRounding="true" />

            </telerik:RadNumericTextBox>

        </EditItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="登录
账号 *" ReadOnly="True" DataField="Account" SortExpression="Account"
UniqueName="Account">

        <ItemTemplate>

            <asp:Label ID="Label2" runat="server" Text='<%#
Eval("Account") %>'></asp:Label>

        </ItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="100" ItemStyle-Width="100" HeaderText="姓
名 *" ReadOnly="True" DataField="RealName" SortExpression="RealName"
UniqueName="RealName">

        <ItemTemplate>

            <asp:Label ID="Label13" runat="server" Text='<%#
Eval("RealName") %>'></asp:Label>

        </ItemTemplate>

    </telerik:GridTemplateColumn>

```

```

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="手机
号码 *" ReadOnly="True" DataField="Phone" SortExpression="Phone"
UniqueName="Phone">

        <ItemTemplate>

            <asp:Label ID="Label15" runat="server" Text='< %#
Eval("Phone") %>'> </asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadTextBox ID="Phone" runat="server"
EnabledStyle-HorizontalAlign="Center" Width="72" MaxLength="11" Text='< %#
Bind("Phone") %>'>

            </telerik:RadTextBox>

        </EditItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60" HeaderText="状态
" DataField="State" SortExpression="State" UniqueName="State">

        <ItemTemplate>

            <asp:Label ID="Label14" runat="server" Text='< %#
Eval("State") %>'> </asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadComboBox ID="State" runat="server"
Width="52" EnableTextSelection="true" Text='< %# Bind("State") %>'>

            <Items>

                <telerik:RadComboBoxItem Text="" Value="-1" />

```



```

        <telerik:RadComboBoxItem Text="启用" Value="1" />

        <telerik:RadComboBoxItem Text="停用" Value="4" />

        <telerik:RadComboBoxItem Text="删除" Value="5" />

    </Items>

</telerik:RadComboBox>

</EditItemTemplate>

</telerik:GridTemplateColumn>

</Columns>

    <PagerStyle Mode="NextPrevAndNumeric"
PageSizes="10,20,50,100" Position="Bottom" PageSizeControlType="RadComboBox"
AlwaysVisible="true" PagerTextFormat="{4} 第{0}页，共{1}页；第{2}-{3}项，共{5}项" />

</MasterTableView>

</telerik:RadGrid>

<div class="ui divider"> </div>

<h6 class="ui red header left floated"><i class="ui red circle
icon"> </i>借调</h6>

<div style="clear: both;"> </div>

    <telerik:RadGrid ID="grid_b" runat="server" AllowPaging="true"
AutoGenerateColumns="false" LocalizationPath=" ../Language" AllowSorting="True"
PageSize="10" GridLines="None" OnNeedDataSource="grid_b_NeedDataSource"
OnBatchEditCommand="grid_b_BatchEditCommand" OnItemCreated="grid_ItemCreated">

        <MasterTableView EditMode="Batch"
DataKeyNames="Id,DepartmentId,Type" CommandItemDisplay="Top"
HorizontalAlign="NotSet" ShowHeader="true" ShowHeadersWhenNoRecords="true"
NoMasterRecordsText="">

            <BatchEditingSettings EditType="Row"
OpenEditingEvent="DbClick" />

```

```

        <CommandItemSettings
ShowAddNewRecordButton="False"> </CommandItemSettings>

        <HeaderStyle HorizontalAlign="Center" />

        <Columns>

            <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="登录
账号 *" ReadOnly="True" DataField="Account" SortExpression="Account"
UniqueName="Account">

                <ItemTemplate>

                    <asp:Label ID="Label2" runat="server" Text='<%#
Eval("Account") %>'> </asp:Label>

                </ItemTemplate>

            </telerik:GridTemplateColumn>

            <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="100" ItemStyle-Width="100" HeaderText="姓
名 *" ReadOnly="True" DataField="RealName" SortExpression="RealName"
UniqueName="RealName">

                <ItemTemplate>

                    <asp:Label ID="Label13" runat="server" Text='<%#
Eval("RealName") %>'> </asp:Label>

                </ItemTemplate>

            </telerik:GridTemplateColumn>

            <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="手机
号码 *" ReadOnly="True" DataField="Phone" SortExpression="Phone"
UniqueName="Phone">

                <ItemTemplate>

```

```

        <asp:Label ID="Label15" runat="server" Text='<##
Eval("Phone") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox ID="Phone" runat="server"
EnabledStyle-HorizontalAlign="Center" Width="72" MaxLength="11" Text='<##
Bind("Phone") %>'>

        </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ReadOnly="true" ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="借调
状态" DataField="Type" SortExpression="Type" UniqueName="Type">

    <ItemTemplate>

        <asp:Label ID="Label14" runat="server" Text='<##
((Homory.Model.DepartmentUserType)Eval("Type")) == Homory.Model.DepartmentUserType.借
调前部门主职教师？"借出"："借入" %>'> </asp:Label>

    </ItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn ItemStyle-HorizontalAlign="Left"
HeaderStyle-Width="300" ItemStyle-Width="300" HeaderText="借调信息" ReadOnly="True"
DataField="Type" SortExpression="Type" UniqueName="Info">

    <ItemTemplate>

        <asp:Label ID="Label16" runat="server" Text='<##
GetInfo(Container.DataItem as Homory.Model.ViewTeacher) %>'> </asp:Label>

    </ItemTemplate>

</telerik:GridTemplateColumn>

```

```

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderText="撤销借调" HeaderStyle-Width="80" ItemStyle-
Width="80" UniqueName="QuitBorrow">

            <ItemTemplate>

                <asp:Label runat="server" Text="保留"> </asp:Label>

            </ItemTemplate>

            <EditItemTemplate>

                <telerik:RadComboBox ID="State" runat="server"
Width="52" EnableTextSelection="true" Text='<%# Bind("State") %>'>

                    <Items>

                        <telerik:RadComboBoxItem Text="保留" Value="1" />

                        <telerik:RadComboBoxItem Text="撤销" Value="5" />

                    </Items>

                </telerik:RadComboBox>

            </EditItemTemplate>

        </telerik:GridTemplateColumn>

    </Columns>

    <PagerStyle Mode="NextPrevAndNumeric"
PageSizes="10,20,50,100" Position="Bottom" PageSizeControlType="RadComboBox"
AlwaysVisible="true" PagerTextFormat="{4} 第{0}页，共{1}页；第{2}-{3}项，共{5}项" />

</MasterTableView>

</telerik:RadGrid>

<div class="ui divider"> </div>

<h6 class="ui black header left floated"> <i class="ui black circle
icon"> </i>可访</h6>

<div style="clear: both;"> </div>

```

```

        <telerik:RadGrid ID="grid_view" runat="server" AllowPaging="true"
AutoGenerateColumns="false" LocalizationPath=" ../Language" AllowSorting="True"
PageSize="10" GridLines="None" OnNeedDataSource="grid_view_NeedDataSource"
OnBatchEditCommand="grid_view_BatchEditCommand" OnItemCreated="grid_ItemCreated">

        <MasterTableView EditMode="Batch" DataKeyNames="Id"
CommandItemDisplay="Top" HorizontalAlign="NotSet" ShowHeader="true"
ShowHeadersWhenNoRecords="true" NoMasterRecordsText="">

        <BatchEditingSettings EditType="Row"
OpenEditingEvent="DbClick" />

        <CommandItemSettings
ShowAddNewRecordButton="False"> </CommandItemSettings>

        <HeaderStyle HorizontalAlign="Center" />

        <Columns>

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="登录
账号 *" ReadOnly="True" DataField="Account" SortExpression="Account"
UniqueName="Account">

        <ItemTemplate>

        <asp:Label ID="Label2" runat="server" Text='<%#
Eval("Account") %>'> </asp:Label>

        </ItemTemplate>

        </telerik:GridTemplateColumn>

        <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="100" ItemStyle-Width="100" HeaderText="姓
名 *" ReadOnly="True" DataField="RealName" SortExpression="RealName"
UniqueName="RealName">

        <ItemTemplate>

        <asp:Label ID="Label13" runat="server" Text='<%#
Eval("RealName") %>'> </asp:Label>

```

```

        </ItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80" ItemStyle-Width="80" HeaderText="手机
号码 *" ReadOnly="True" DataField="Phone" SortExpression="Phone"
UniqueName="Phone">

        <ItemTemplate>

            <asp:Label ID="Label15" runat="server" Text='<%#
Eval("Phone") %>'></asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadTextBox ID="Phone" runat="server"
EnabledStyle-HorizontalAlign="Center" Width="72" MaxLength="11" Text='<%#
Bind("Phone") %>'>

        </telerik:RadTextBox>

        </EditItemTemplate>

    </telerik:GridTemplateColumn>

    <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="60" ItemStyle-Width="60" HeaderText="状态
" DataField="State" SortExpression="State" UniqueName="State">

        <ItemTemplate>

            <asp:Label ID="Label14" runat="server" Text='<%#
Eval("State") %>'></asp:Label>

        </ItemTemplate>

        <EditItemTemplate>

            <telerik:RadComboBox ID="State" runat="server"
Width="52" EnableTextSelection="true" Text='<%# Bind("State") %>'>

```

```

        <Items>

            <telerik:RadComboBoxItem Text="" Value="-1" />

            <telerik:RadComboBoxItem Text="启用" Value="1" />

            <telerik:RadComboBoxItem Text="停用" Value="4" />

            <telerik:RadComboBoxItem Text="删除" Value="5" />

        </Items>

    </telerik:RadComboBox>

</EditItemTemplate>

</telerik:GridTemplateColumn>

</Columns>

    <PagerStyle Mode="NextPrevAndNumeric"
PageSizes="10,20,50,100" Position="Bottom" PageSizeControlType="RadComboBox"
AlwaysVisible="true" PagerTextFormat="{4} 第{0}页，共{1}页；第{2}-{3}项，共{5}项" />

</MasterTableView>

</telerik:RadGrid>

</div>

</td>

</tr>

</table>

</div>

</telerik:RadAjaxPanel>

</form>

</body>

</html>

```

4. 全局消息通知

//调用 JS 函数弹出提示信息

```
protected void Notify(RadAjaxPanel panel, string message, string type)

{

    panel.ResponseScripts.Add(string.Format("notify(null, '{0}', '{1}');", message, type));

}
```

5. 菜单项加载

//加载顶级菜单

```
protected void LoadInit()

{

    _menus =

        HomoryContext.Value.Menu.Where(o => o.ApplicationId ==

HomoryCoreConstant.ApplicationKey && o.State == State.启用)

        .ToList();

    var menuList = _menus.Where(o => o.ParentId == null)

        .OrderBy(o => o.Ordinal)

        .ToList();

    repeater.DataSource = menuList;

    repeater.DataBind();

}
```

//加载子菜单

```
protected string SubMenu(Menu menu)

{

    var query = _menus.Where(o => o.ParentId == menu.Id).OrderBy(o =>

o.Ordinal).ToList();

}
```



```

var sb = new StringBuilder();

foreach (var item in query)
{
    if (CurrentRights.Contains(item.RightName))

        sb.Append(string.Format("<a class=\"item\" href=\"{0}\">{1}<i
class=\"icon\"></i> </a>", item.Redirect.StartsWith("+") ?
item.Redirect.Substring(1).FromHomoryConfig() : item.Redirect, item.Name));
}

return sb.ToString();
}

<link href=" ../Content/Core/css/sidebar.css" rel="stylesheet" />

<script src=" ../Content/Core/js/sidebar.js"> </script>

<telerik:RadScriptManager ID="sm" runat="server"> </telerik:RadScriptManager>

<div class="ui vertical labeled icon sidebar menu" id="sidebar">

    <div class="item">

        <a href=" ../Go/Home.aspx"> <i id="sideHome" class="circular large home
icon"> </i>

            <br />

            <br />

            <b>基础平台</b> </a>

    </div>

    <asp:Repeater ID="repeater" runat="server">

        <ItemTemplate>

```

```

        <div class="item">

            <div class='<## string.Format(" ui small circular button {0}",
Eval("Icon")) %>' data-id='<## Eval("Id") %>'><## Eval("Name") %></div>

            <div class='<## string.Format("sub menu {0}", Eval("Icon")) %>'
data-id='<## Eval("Id") %>'><## SubMenu(Container.DataItem as Homory.Model.Menu)
%></div>

        </div>

    </ItemTemplate>

</asp:Repeater>

</div>

<div class="container">

    <a id="sidebarSwitcher" class="launch item rootPointer" title="菜单">

        <i class="icon list layout large"></i><a id="u" runat="server"
onclick="nameClick(); return false;" class="welcome
rootPointer"></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a id="qb" runat="server" class="welcome
rootPointer" onserverclick="qb_Click">退出</a>

    </a>

</div>

<div class="ui center aligned page grid" style="margin:0;padding:0;">

    <div class="column">

        <div id="bar" class="ui inverted black center aligned segment">

            <h2 id="headInfo" class="ui big header"></h2>

        </div>

    </div>

</div>

</div>

```

6. 权限管理

//判断用户对部门的可访问性和可操作性

```
protected void view_OnNeedDataSource(object sender,
RadListViewNeedDataSourceEventArgs e)

{

    var departmentId = tree.SelectedNode == null || tree.SelectedNode.Level == 0

        ? (Guid?)null

        : Guid.Parse(tree.SelectedNode.Value);

    var list = departmentId.HasValue

        ? HomoryContext.Value.ViewTeacher.Where(o => o.DepartmentId ==
departmentId.Value && o.State < State.审核 && (o.Type == DepartmentUserType.部门主职教师 || o.Type == DepartmentUserType.部门兼职教师 || o.Type == DepartmentUserType.借调后
部门主职教师)).ToList()

        : null;

    view.Visible = departmentId.HasValue;

    var stext = peek.Text.Trim();

    view.DataSource = departmentId.HasValue

        ? list.OrderBy(o => o.Type).ThenBy(o => o.PriorOrdinal).ThenBy(o =>
o.RealName).ToList().Where(o => o.Account.Contains(stext) || o.RealName.Contains(stext) ||
o.Phone.Contains(stext) || o.IDCard.Contains(stext)).ToList()

        : null;

}
```

//人员加入某个权限组

```
protected void view_OnItemDrop(object sender, RadListViewItemDragDropEventArgs e)

{

    try
```

```

    {
        var target = e.DestinationHtmlElement;

        var roleId = Guid.Parse(target);

        var userId = Guid.Parse(e.DraggedItem.GetDataKeyValue("Id").ToString());

        var ur = new UserRole
        {
            UserId = userId,
            RoleId = roleId,
            State = State.启用
        };

        HomoryContext.Value.UserRole.AddOrUpdate(ur);

        HomoryContext.Value.SaveChanges();

        LogOp(OperationType.编辑);

        grid.Rebind();

        Notify(panel, "操作成功", "success");
    }

    // ReSharper disable EmptyGeneralCatchClause
    catch

    // ReSharper restore EmptyGeneralCatchClause
    {
    }

    }

    protected void view_OnItemDrop(object sender, RadListViewItemDragDropEventArgs e)
    {

```

```

try
{
    var target = e.DestinationHtmlElement;

    var roleId = Guid.Parse(target);

    var userId = Guid.Parse(e.DraggedItem.GetDataKeyValue("Id").ToString());

    var ur = new UserRole
    {
        UserId = userId,
        RoleId = roleId,
        State = State.启用
    };

    HomoryContext.Value.UserRole.AddOrUpdate(ur);

    HomoryContext.Value.SaveChanges();

    LogOp(OperationType.编辑);

    grid.Rebind();

    Notify(panel, "操作成功", "success");
}

// ReSharper disable EmptyGeneralCatchClause
catch

// ReSharper restore EmptyGeneralCatchClause
{
}

}

//获取当前应用程序策略

```

```
public List<ApplicationPolicy> Policies
```

```
{  
    get  
    {  
        if (_policies == null || _policies.Count == 0)  
        {  
            _policies = HomoryContext.Value.ApplicationPolicy.ToList();  
        }  
        return _policies;  
    }  
}
```

```
<form id="formHome" runat="server">
```

```
    <div>
```

```
        <homory:SideBar runat="server" ID="SideBar" />
```

```
    </div>
```

```
    <telerik:RadAjaxLoadingPanel ID="loading" runat="server">
```

```
        <i class="ui huge teal loading icon" style="margin-top: 50px;"> </i>
```

```
        <div>&nbsp;</div>
```

```
        <div style="color: #564F8A; font-size: 16px;">正在加载 请稍候....</div>
```

```
    </telerik:RadAjaxLoadingPanel>
```

```
    <telerik:RadAjaxPanel ID="panel" runat="server" CssClass="ui left aligned stackable  
page grid" Style="margin: 0; padding: 0;" LoadingPanelID="loading">
```

```
        <div class="ui row"> </div>
```

```
        <div class="row">
```

```
<div class="eight wide column">  
  
    <h6 class="ui teal header"><i class="ui teal circle icon"></i>教师</h6>  
  
    <telerik:RadComboBox ID="combo" runat="server" AutoPostBack="true"  
OnSelectedIndexChanged="combo_SelectedIndexChanged" DataTextField="Name"  
DataValueField="Id" Label="选择学校：" Width="220px" Filter="Contains"  
MarkFirstMatch="true" AllowCustomText="true" Height="202px">  
  
        <ItemTemplate>  
  
            <%# Eval("Name") %><!--<%# CountChildren(Container.DataItem as  
Department) %>--%>  
  
        </ItemTemplate>  
  
    </telerik:RadComboBox>  
  
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
  
    <telerik:RadSearchBox ID="peek" runat="server" OnSearch="peek_Search"  
EmptyMessage="查找..." EnableAutoComplete="false">  
  
    </telerik:RadSearchBox>  
  
  
    <div class="eight wide column">  
  
        <h6 class="ui purple header"><i class="ui purple circle icon"></i>角色</h6>  
  
    </div>  
  
  
    <div class="row">  
  
        <div class="eight wide left aligned column coreLeft">  
  
            <table>  
  
                <tr class="coreTop">  
  
                    <td>
```

```

        <telerik:RadTreeView ID="tree" runat="server" Width="250"
DataTextField="Name" DataValueField="Id" DataFieldID="Id" DataFieldParentID="ParentId"
OnNodeClick="tree_NodeClick" EnableDragAndDrop="False">

```

```

        <NodeTemplate>

```

```

            <%# ForceTreeName(Container.DataItem as
Homory.Model.Department) %>

```

```

        </NodeTemplate>

```

```

    </telerik:RadTreeView>

```

```

</td>

```

```

<td>

```

```

        <telerik:RadListView ID="view" runat="server" DataKeyNames="Id"
ClientDataKeyNames="Id" OnNeedDataSource="view_OnNeedDataSource"
OnItemDrop="view_OnItemDrop">

```

```

        <ItemTemplate>

```

```

            <!-- ReSharper disable UnknownCssClass --%>

```

```

            <div id='<%# Eval("Id") %>' class="rlvI RadTreeView_Default
rlvDrag rootPointer ui basic segment left floated"
onmousedown="Telerik.Web.UI.RadListView.HandleDrag(event, '<%#
Container.OwnerListView.ClientID %>', <%# Container.DisplayIndex%>);">

```

```

            <!-- ReSharper restore UnknownCssClass --%>

```

```

            <i class="ui teal circle icon"></i>&nbsp;<%# Eval("RealName")
%>

```

```

        </div>

```

```

    </ItemTemplate>

```

```

    <ClientSettings AllowItemsDragDrop="true"></ClientSettings>

```

```

</telerik:RadListView>

```

```

</td>

```



```

        </tr>

    </table>

</div>

<div class="eight wide left aligned column coreLeft">

    <table>

        <tr class="coreTop">

            <td>

                <telerik:RadListView ID="viewX" runat="server" DataKeyNames="Id"
ClientDataKeyNames="Id" OnNeedDataSource="viewX_OnNeedDataSource"
OnItemDrop="viewX_OnItemDrop">

                    <ItemTemplate>

                        <!-- ReSharper disable UnknownCssClass --%>

                        <div id='<%=# Eval("Id") %>' class="rlvI RadTreeView_Default
rlvDrag rootPointer ui basic segment left floated"
onmousedown="Telerik.Web.UI.RadListView.HandleDrag(event, '<%=#
Container.OwnerListView.ClientID %>', <%=# Container.DisplayIndex%>);">

                            <!-- ReSharper restore UnknownCssClass --%>

                            <i class="ui purple circle icon"> </i> &nbsp;<%=# Eval("Name") %>

                        </div>

                    </ItemTemplate>

                    <ClientSettings AllowItemsDragDrop="true"> </ClientSettings>

                </telerik:RadListView>

            </td>

        </tr>

    <tr>

```

```

        <td>

        <div class="ui divider"> </div>

        <h6 class="ui red header"> <i class="ui red circle icon"> </i>授权列表
    </h6>

    <telerik:RadGrid ID="grid" runat="server" AllowPaging="True"
AutoGenerateColumns="false" LocalizationPath="../Language" AllowSorting="True"
GridLines="None" OnNeedDataSource="grid_OnNeedDataSource"
OnDeleteCommand="grid_OnDeleteCommand">

        <MasterTableView DataKeyNames="UserId,RoleId"
CommandItemDisplay="None" HorizontalAlign="NotSet" ShowHeader="true"
ShowHeadersWhenNoRecords="true" NoMasterRecordsText="">

            <HeaderStyle HorizontalAlign="Center" />

            <ItemStyle HorizontalAlign="Center"> </ItemStyle>

            <AlternatingItemStyle
HorizontalAlign="Center"> </AlternatingItemStyle>

            <Columns>

                <telerik:GridBoundColumn HeaderText="教师"
DataField="UserName" SortExpression="UserName" UniqueName="UserName">

                    </telerik:GridBoundColumn>

                <telerik:GridBoundColumn HeaderText="角色"
DataField="RoleName" SortExpression="RoleName" UniqueName="RoleName">

                    </telerik:GridBoundColumn>

                <telerik:GridButtonColumn HeaderText=""
CommandName="Delete" ButtonType="ImageButton" />

            </Columns>

            <PagerStyle Mode="NextPrevAndNumeric"
PageSizes="10,20,50,100" Position="Bottom" PageSizeControlType="RadComboBox"
AlwaysVisible="true" PagerTextFormat="{4} 第{0}页，共{1}页；第{2}-{3}项，共{5}项" />

```

```

        </MasterTableView>

        </telerik:RadGrid>

    </td>

</tr>

</table>

</div>

</div>

</telerik:RadAjaxPanel>

</form>

```

7. 组织管理

//添加部门

```

HomoryContext.Value.Department.Add(new Homory.Model.Department
{
    Id = nid,
    Name = name,
    TopId = nid,
    DisplayName = name,
    Level = 0,
    Hidden = false,
    Ordinal = ordinal,
    State = state,
    Code = code,
    BuildType = buildType,
    ClassType = classType,

```

```

Type = DepartmentType.学校

});

try { DepartmentHelper.InsertCampus(nid.ToString().ToUpper(), name, ordinal); } catch {
Notify(panel, "办公平台学校新增失败", "warn"); }

HomoryContext.Value.SaveChanges();

//自动生成年级数据

protected void AutoGenerateGrades()

{

var parentId = tree.SelectedNode == null || tree.SelectedNode.Level != 0 ? (Guid?)(null) :
Guid.Parse(tree.SelectedNode.Value);

if (!parentId.HasValue) return;

ClassType classType = HomoryContext.Value.Department.First(o => o.Id ==
parentId.Value).ClassType;

var startYear = __Year - GradeCount(classType) + 1;

for (var i = 0; i < GradeCount(classType); i++)

{

var j = startYear + i;

var ding = j + GradeCount(classType);

if (HomoryContext.Value.Department.Count(o => o.Type == DepartmentType.班级 &&
o.ParentId == parentId.Value && o.Ordinal == ding) == 0)

{

HomoryContext.Value.Department.Add(new Homory.Model.Department

{

Id = HomoryContext.Value.GetId(),

Name = string.Format("{0}届", ding),

```

```

DisplayName = string.Format("{0}届", ding),
ParentId = parentId,
TopId = parentId.Value,
Level = 1,
Hidden = false,
Ordinal = ding,
State = State.启用,
Type = DepartmentType.班级,
ClassType = ClassType.无,
BuildType = BuildType.无,
Code = string.Empty
});
LogOp(OperationType.新增);
}
else
{
HomoryContext.Value.Department.Where(o => o.Type == DepartmentType.班级 &&
o.ParentId == parentId.Value && o.Ordinal == ding).Update(o => new
Homory.Model.Department { State = State.启用 });
}
}

HomoryContext.Value.Department.Where(o => o.Type == DepartmentType.班级 &&
o.ParentId == parentId.Value && o.Ordinal <= __Year).Update(
o => new Homory.Model.Department { State = State.删除 });

HomoryContext.Value.SaveChanges();

```

```

RebindBatch();

}

//加载班级班主任

protected void LoadCharging()
{
    if (tree.SelectedNode.Level != 2) return;

    var classId = Guid.Parse(tree.SelectedNode.Value);

    var du = HomoryContext.Value.DepartmentUser.SingleOrDefault(o => o.DepartmentId ==
classId && o.State == State.启用 && o.Type == DepartmentUserType.班级班主任);

    if (du == null)
    {
        charging.Visible = false;

        charging.Text = string.Empty;

        charging.CommandArgument = string.Empty;
    }
    else
    {
        charging.Visible = true;

        var user = du.User;

        charging.Text = user.RealName;

        charging.CommandArgument = user.Id.ToString();
    }
}

<form id="formHome" runat="server">

```

```

<div>

    <homory:SideBar runat="server" ID="SideBar" />

</div>

<telerik:RadAjaxLoadingPanel ID="loading" runat="server">

    <i class="ui huge teal loading icon" style="margin-top: 50px;"> </i>

    <div>&nbsp;</div>

    <div style="color: #564F8A; font-size: 16px;">正在加载 请稍候....</div>

</telerik:RadAjaxLoadingPanel>

    <telerik:RadAjaxPanel ID="panel" runat="server" CssClass="ui center aligned
page grid" style="margin:0;padding:0;" LoadingPanelID="loading">

        <div class="column">

            <telerik:RadGrid ID="grid" runat="server"
AutoGenerateColumns="false" LocalizationPath=" ../Language" AllowSorting="True"
PageSize="20" GridLines="None" OnNeedDataSource="grid_NeedDataSource"
OnBatchEditCommand="grid_BatchEditCommand" OnItemCreated="grid_ItemCreated">

                <MasterTableView EditMode="Batch"
DataKeyNames="Id" CommandItemDisplay="Top" HorizontalAlign="NotSet"
ShowHeader="true" ShowHeadersWhenNoRecords="true" NoMasterRecordsText="">

                    <BatchEditingSettings EditType="Row"

OpenEditingEvent="DbClick" />

                    <HeaderStyle HorizontalAlign="Center" />

                    <Columns>

                        <telerik:GridTemplateColumn ItemStyle-
Width="80" HeaderStyle-Width="80" HeaderText="序号" DataField="Ordinal"
SortExpression="Ordinal" UniqueName="Ordinal">

                            <ItemTemplate>

```

```

                                <asp:Label runat="server"
Text='<%= Eval("Ordinal") %>'> </asp:Label>

                                </ItemTemplate>

                                <EditItemTemplate>

                                <telerik:RadNumericTextBox
ID="Ordinal" runat="server" EnabledStyle-HorizontalAlign="Center" Width="64"
MinValue="1" MaxValue="999999" AllowOutOfRangeAutoCorrect="true" Value='<%=
Bind("Ordinal") %>'>

                                <NumberFormat

DecimalDigits="0" AllowRounding="true" />

                                </telerik:RadNumericTextBox>

                                </EditItemTemplate>

                                </telerik:GridTemplateColumn>

                                <telerik:GridTemplateColumn ItemStyle-
HorizontalAlign="Left" ItemStyle-Width="250" HeaderStyle-Width="250" HeaderText="机构
名称 *" DataField="Name" SortExpression="Name" UniqueName="Name">

                                <ItemTemplate>

                                <asp:Label runat="server"

Text='<%= Eval("Name") %>'> </asp:Label>

                                </ItemTemplate>

                                <EditItemTemplate>

                                <telerik:RadTextBox

ID="Name" runat="server" EnabledStyle-HorizontalAlign="Center" Width="64"
MaxLength="16" Text='<%= Bind("Name") %>'>

                                </telerik:RadTextBox>

                                </EditItemTemplate>

```



```

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn
HeaderText="机构代码" DataField="Code" SortExpression="Code" UniqueName="Code">

    <ItemTemplate>

        <asp:Label runat="server"
Text='<%=# Eval("Code") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadTextBox
ID="Code" runat="server" EnabledStyle-HorizontalAlign="Center" Width="64"
MaxLength="32" Text='<%=# Bind("Code") %>'>

    </telerik:RadTextBox>

    </EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn HeaderText="学校类别"
DataField="ClassType" SortExpression="ClassType" UniqueName="ClassType">

    <ItemTemplate>

        <asp:Label
ID="classTypeLabel" runat="server" Text='<%=# Eval("ClassType") %>'> </asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadComboBox
ID="ClassType" runat="server" Width="90" EnableTextSelection="true" Text='<%=#
Eval("ClassType") %>'>

        <Items>

        <telerik:RadComboBoxItem Text="" Value="0" />

```

```

        <telerik:RadComboBoxItem Text="九年一贯制" Value="1" />
        <telerik:RadComboBoxItem Text="幼儿园" Value="2" />
        <telerik:RadComboBoxItem Text="小学" Value="3" />
        <telerik:RadComboBoxItem Text="初中" Value="4" />
        <telerik:RadComboBoxItem Text="其他" Value="5" />

        </Items>

    </telerik:RadComboBox>

</EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn HeaderText="学校办别"
DataField="BuildType" SortExpression="BuildType" UniqueName="BuildType">

    <ItemTemplate>

        <asp:Label
ID="buildTypeLabel" runat="server" Text='<%# Eval("BuildType") %>'></asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadComboBox
ID="BuildType" runat="server" Width="140" EnableTextSelection="true" Text='<%#
Eval("BuildType") %>'>

        <Items>

            <telerik:RadComboBoxItem Text="" Value="0" />

            <telerik:RadComboBoxItem Text="教育部门社会集体办" Value="1" />

            <telerik:RadComboBoxItem Text="民办" Value="2" />

```

```

</Items>

</telerik:RadComboBox>

</EditItemTemplate>

</telerik:GridTemplateColumn>

<telerik:GridTemplateColumn
HeaderText="状态" DataField="State" SortExpression="State" UniqueName="State">

    <ItemTemplate>

        <asp:Label ID="stateLabel"
runat="server" Text='<%# Eval("State") %>'></asp:Label>

    </ItemTemplate>

    <EditItemTemplate>

        <telerik:RadComboBox
ID="State" runat="server" Width="64" EnableTextSelection="true" Text='<%# Eval("State")
%>'>

            <Items>

                <telerik:RadComboBoxItem Text="" Value="-1" />

                <telerik:RadComboBoxItem Text="启用" Value="1" />

                <telerik:RadComboBoxItem Text="停用" Value="4" />

                <telerik:RadComboBoxItem Text="删除" Value="5" />

            </Items>

        </telerik:RadComboBox>

    </EditItemTemplate>

```

```

        </telerik:GridTemplateColumn>

    </Columns>

</MasterTableView>

</telerik:RadGrid>

</div>

</telerik:RadAjaxPanel>

</form>

```

8. 课程管理

//添加课程

```

public bool CourseAdd(Entities db, string name, State state, int ordinal, bool range)
{
    try
    {
        var course = new Catalog
        {
            Id = db.GetId(),
            ParentId = HomoryCoreConstant.CourseOtherId,
            TopId = range ? Guid.Empty : (Guid?)null,
            Name = name,
            State = state,
            Ordinal = ordinal,
            Type = CatalogType.课程
        };
    }
}

```

```

var ex = db.Catalog.SingleOrDefault(o => o.Name == name && o.Type ==
CatalogType.课程);

if (ex == null)
{
    db.Catalog.Add(course);

    db.SaveChanges();
}
else
{
    if (ex.State > State.内置)
    {
        ex.State = state;

        ex.Ordinal = ordinal;

        db.SaveChanges();
    }
}

return true;
}

catch
{
    return false;
}
}

//设定班级所学课程

```

```

public bool CourseLearned(Entities db, Guid classId, Guid courseId)
{
    try
    {
        if (db.Learned.Count(o => o.DepartmentId == classId && o.CourseId == courseId) > 0)
        {
            var learned = db.Learned.First(o => o.DepartmentId == classId && o.CourseId == courseId);
            learned.State = learned.State == State.启用 ? State.删除 : State.启用;
            LogOp(learned.State);
        }
        else
        {
            var learned = new Learned
            {
                DepartmentId = classId,
                CourseId = courseId,
                State = State.启用
            };
            db.Learned.Add(learned);
            LogOp(OperationType.新增);
        }
        db.SaveChanges();
        return true;
    }
}

```

```

catch
{
return false;
}
}

<telerik:RadAjaxLoadingPanel ID="loading" runat="server">

    <i class="ui huge teal loading icon" style="margin-top: 50px;"> </i>

    <div>&nbsp;</div>

    <div style="color: #564F8A; font-size: 16px;">正在加载 请稍候....</div>

</telerik:RadAjaxLoadingPanel>

<telerik:RadAjaxPanel ID="panel" runat="server" CssClass="ui left aligned stackable
page grid" Style="margin: 0; padding: 0;" LoadingPanelID="loading">

    <div class="ui row"> </div>

    <div class="row">

        <div class="eight wide column">

            <h6 class="ui teal header"> <i class="ui teal circle icon"> </i>教师</h6>

            <telerik:RadComboBox ID="combo" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="combo_SelectedIndexChanged" DataTextField="Name"
DataValueField="Id" Label="选择学校 : " Width="220px" Filter="Contains"
MarkFirstMatch="true" AllowCustomText="true" Height="202px">

                <ItemTemplate>

                    <%# Eval("Name") %> <!--<%# CountChildren(Container.DataItem as
Department) %>-->

                </ItemTemplate>

            </telerik:RadComboBox>

```

```
<div class="ui purple header"><i>班级课程</i></div>
```

```
<telerik:RadSearchBox ID="peek" runat="server" OnSearch="peek_Search" EmptyMessage="查找...." EnableAutoComplete="false">
```

```
</telerik:RadSearchBox>
```

```
</div>
```

```
<div class="eight wide column">
```

```
<h6 class="ui purple header"><i>班级课程</i></h6>
```

```
<telerik:RadComboBox ID="comboX" runat="server" AutoPostBack="true" OnSelectedIndexChanged="comboX_SelectedIndexChanged" DataTextField="Name" DataValueField="Id" Label="选择学校：" Width="220px" Filter="Contains" MarkFirstMatch="true" AllowCustomText="true" Height="202px">
```

```
<ItemTemplate>
```

```
<%# Eval("Name") %><!--<%# CountChildren(Container.DataItem as Department) %>--%>
```

```
</ItemTemplate>
```

```
</telerik:RadComboBox>
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
<div class="eight wide column">
```

```
<table>
```

```
<tr class="coreTop">
```

```
<td>
```

```
<telerik:RadTreeView ID="tree" runat="server" Width="250" DataTextField="Name" DataValueField="Id" DataFieldID="Id" DataFieldParentID="ParentId" OnNodeClick="tree_NodeClick" EnableDragAndDrop="False">
```



```

        <NodeTemplate>

            <%# Eval("Name").ToString() %>

        </NodeTemplate>

    </telerik:RadTreeView>

</td>

<td>&nbsp;</td>

<td>

    <telerik:RadListView ID="view" runat="server" DataKeyNames="Id"
ClientDataKeyNames="Id" OnNeedDataSource="view_OnNeedDataSource"
OnItemDrop="view_OnItemDrop">

        <ItemTemplate>

            <%-- ReSharper disable UnknownCssClass --%>

            <div id='<%# Eval("Id") %>' class="rlvI RadTreeView_Default
rlvDrag rootPointer ui basic segment left floated"
onmousedown="Telerik.Web.UI.RadListView.HandleDrag(event, '<%#
Container.OwnerListView.ClientID %>', <%# Container.DisplayIndex%>);">

                <%-- ReSharper restore UnknownCssClass --%>

                <i class='<%# GetIcon(Container.DataItem as
Homory.Model.ViewTeacher) %>'></i>&nbsp;<%# Eval("RealName") %>

            </div>

        </ItemTemplate>

    <ClientSettings AllowItemsDragDrop="true"></ClientSettings>

</telerik:RadListView>

</td>

</tr>

</table>

```

```

</div>

<div class="eight wide column">

    <table>

        <tr class="coreTop">

            <td>

                <telerik:RadTreeView ID="treeX" runat="server" Width="250"
DataTextField="Name" DataValueField="Id" DataFieldID="Id" DataFieldParentID="ParentId"
OnNodeClick="treeX_NodeClick" EnableDragAndDrop="False">

                    <NodeTemplate>

                        <%#
GenerateTreeName((Homory.Model.Department)Container.DataItem, Container.Index,
Container.Level) %>

                    </NodeTemplate>

                </telerik:RadTreeView>

            </td>

            <td>&nbsp;</td>

            <td>

                <telerik:RadListView ID="viewX" runat="server"
DataKeyNames="DepartmentId,CourseId" ClientDataKeyNames="DepartmentId,CourseId"
OnNeedDataSource="viewX_OnNeedDataSource" OnItemDrop="viewX_OnItemDrop">

                    <ItemTemplate>

                        <%-- ReSharper disable UnknownCssClass --%>

                        <div id='<%# Eval("CourseId") %>' style="line-height: 30px;
vertical-align: middle;" class="rlvI RadTreeView_Default rlvDrag rootPointer ui basic
segment" onmousedown="Telerik.Web.UI.RadListView.HandleDrag(event, '<%#
Container.OwnerListView.ClientID %>', <%# Container.DisplayIndex%>);">

                            <%-- ReSharper restore UnknownCssClass --%>

```

```

        <i class="ui purple circle icon"></i>&nbsp;< %#
Eval("CourseName") %>&nbsp;< %# GetTeacher(Guid.Parse(treeX.SelectedNode.Value),
(Guid)Eval("CourseId")) %>&nbsp;<asp:ImageButton ID="toDel" runat="server"
Visible='< %# HasTeacher(Guid.Parse(treeX.SelectedNode.Value), (Guid)Eval("CourseId")) %>'
ImageUrl="~/Images/Common/Delete.png" CommandArgument='< %# Eval("CourseId")
%>' OnClick="toDel_Click" />

    </div>

</ItemTemplate>

<ClientSettings AllowItemsDragDrop="true"></ClientSettings>

</telerik:RadListView>

</td>

</tr>

</table>

</div>

</div>

</telerik:RadAjaxPanel>

```

9. 人员管理

//添加学生

```

public bool StudentAdd(Entities db, Guid campusId, Guid classId, int ordinal, string name,
string account, string passwordInitial, State state, string uniqueId, string idCard, bool?
gender, DateTime? birthday, string nationality, string birthplace, string address, string
charger, string chargerContact)

```

```
{
```

```
try
```

```
{
```

```
string key, salt;
```

```
var password = HomoryCryptor.Encrypt(passwordInitial, out key, out salt);
```

```

var user = new User
{
    Id = db.GetId(),
    Account = account,
    RealName = name,
    DisplayName = name,
    Stamp = Guid.NewGuid(),
    Type = UserType.学生,
    Password = password,
    PasswordEx = null,
    CryptoKey = key,
    CryptoSalt = salt,
    Icon = "~/Common/默认/用户.png",
    State = state,
    Ordinal = ordinal,
    Description = null
};

```

//添加账号以及学生班级关系

```

var userStudent = new Homory.Model.Student
{
    Id = user.Id,
    UniqueId = uniqueId,
    Gender = gender,
    Birthday = birthday,

```

```

Birthplace = birthplace,
Address = address,
Nationality = nationality,
IDCard = idCard,
Charger = charger,
ChargerContact = chargerContact
};

var relation = new DepartmentUser
{
    DepartmentId = classId,
    UserId = user.Id,
    TopDepartmentId = campusId,
    Type = DepartmentUserType.班级学生,
    State = State.启用,
    Ordinal = 0
};

db.User.Add(user);
db.Student.Add(userStudent);
db.DepartmentUser.Add(relation);

return true;
}

catch

{
    return false;
}

```

```
}  
  
}
```

10. 云资源平台基础数据相关

//加载云资源平台操作所得的荣誉分值

```
count1.Value = int.Parse(HomoryContext.Value.Dictionary.Single(o => o.Key ==  
"CreditPublish").Value);
```

```
count2.Value = int.Parse(HomoryContext.Value.Dictionary.Single(o => o.Key ==  
"CreditComment").Value);
```

```
count3.Value = int.Parse(HomoryContext.Value.Dictionary.Single(o => o.Key ==  
"CreditReply").Value);
```

```
count4.Value = int.Parse(HomoryContext.Value.Dictionary.Single(o => o.Key ==  
"CreditRate").Value);
```

```
count5.Value = int.Parse(HomoryContext.Value.Dictionary.Single(o => o.Key ==  
"CreditFavourite").Value);
```

//绑定审计图数据源

```
protected void BindChartDataSource(List<ViewQueryResource> source)
```

```
{
```

```
this.UnDisplayTitle();
```

```
this.ClearSeries();
```

```
this.BuildSeries();
```

```
Dictionary<Guid, string> l = new Dictionary<Guid, string>();
```

```
foreach (var s in HomoryContext.Value.Department.Where(o => o.State == State.启用 &&  
o.Type == DepartmentType.学校).OrderBy(o => o.Ordinal).Select(o => new { o.Id, o.Name  
}).ToList())
```

```
{
```

```
if (CurrentRights.Contains(HomoryCoreConstant.RightGlobal))
```

```

{
    l.Add(s.Id, s.Name);
}

else

{
    var campus = CurrentCampus.Id;

    if (s.Id == campus)

    l.Add(s.Id, s.Name);

}

}

if (l.Count() > 0)

{
    List<Calced> l0 = new List<Calced>();

    List<Calced> l1 = new List<Calced>();

    var width = 0;

    foreach (var s in l)

    {

        l0.Add(new Calced(s.Value, source.Where(o => o.CampusId == s.Key).Sum(o => o.下载资源
        + o.发布文章 + o.发布视频 + o.发布试卷 + o.发布课件 + o.回复评论 + o.收藏资源 + o.浏览资源 +
        o.评定资源 + o.评论资源)));

        l1.Add(new Calced(s.Value, source.Where(o => o.CampusId == s.Key).Sum(o => o.获得积分)));

        width++;

    }

    var array = new List<Calced>[] { l0, l1 };
}

```

```

for (int _index = 0; _index < array.Length; _index++)
{
    foreach (Calced item in array[_index])
    {
        this.BuildSeriesItem(this.c.Series[SeriesIndexes[_index]], (double)item.Num,
            item.Num.ToString());
    }
}

this.c.Width = 100 + 80 * width;

this.c.PlotArea.XAxis.AutoScale = false;

this.c.PlotArea.XAxis.Clear();

this.c.PlotArea.XAxis.AddRange(1, l.Count, 1);

for (int i = 0; i < l.Count; i++)
{
    this.c.PlotArea.XAxis[i].TextBlock.Appearance.AutoTextWrap = AutoTextWrap.True;

    this.c.PlotArea.XAxis[i].TextBlock.Appearance.TextProperties.Color = (i % 2) == 0 ?
        Color.Silver : Color.YellowGreen;

    this.c.PlotArea.XAxis[i].TextBlock.Text = l.ElementAt(i).Value;
}
}

protected static string[] SeriesNames = new string[] { "互动量", "荣誉值" };

protected static int[] SeriesIndexes = new int[] { 0, 1 };

```


//绑定审计表格数据源

```
protected void grid_OnNeedDataSource(object sender, GridNeedDataSourceEventArgs e)
{
    if (combo.SelectedIndex < 0 || comboX.SelectedIndex < 0)
    {
        grid.DataSource = null;
        return;
    }

    var year = int.Parse(combo.SelectedItem.Value);
    var month = int.Parse(comboX.SelectedItem.Value);

    List<ViewQueryResource> query;

    if (CurrentRights.Contains(HomoryCoreConstant.RightGlobal))
    {
        query = month == 0
            ? HomoryContext.Value.ViewQueryResource.Where(o => o.年份 == year).ToList()
            : HomoryContext.Value.ViewQueryResource.Where(o => o.年份 == year && o.月份 ==
                month).ToList();
    }
    else
    {
        var cid = CurrentCampus == null ? (Guid?)null : CurrentCampus.Id;

        if (cid == null)
        {
            query = new List<ViewQueryResource>();
        }
    }
}
```

```

}

else

{

    query = month == 0

    ? HomoryContext.Value.ViewQueryResource.Where(o => o.年份 == year && o.CampusId ==
    CurrentCampus.Id).ToList()

    : HomoryContext.Value.ViewQueryResource.Where(o => o.年份 == year && o.月份 ==
    month && o.CampusId == CurrentCampus.Id).ToList();

}

}

var list = new List<ViewQueryResource>();

foreach (var g in query.GroupBy(o => o.Id))

{

    var obj = new ViewQueryResource();

    obj.学校 = g.First().学校;

    obj.教师 = g.First().教师;

    obj.CampusId = g.First().CampusId;

    obj.Id = g.First().Id;

    obj.下载资源 = g.Sum(o => o.下载资源);

    obj.发布文章 = g.Sum(o => o.发布文章);

    obj.发布视频 = g.Sum(o => o.发布视频);

    obj.发布试卷 = g.Sum(o => o.发布试卷);

    obj.发布课件 = g.Sum(o => o.发布课件);

    obj.回复评论 = g.Sum(o => o.回复评论);

```

```

obj.收藏资源 = g.Sum(o => o.收藏资源);
obj.浏览资源 = g.Sum(o => o.浏览资源);
obj.获得积分 = g.Sum(o => o.获得积分);
obj.评定资源 = g.Sum(o => o.评定资源);
obj.评论资源 = g.Sum(o => o.评论资源);
list.Add(obj);
}

grid.DataSource = list;

if (CurrentRights.Contains(HomoryCoreConstant.RightGlobal))
{
    BindChartDataSource(list);
}
else
{
    BindChartDataSource(list);
}
}

```

11. 数据上传导入

//学生数据上传并导入

```

protected void im_up_FileUploaded(object sender, Telerik.Web.UI.FileUploadedEventArgs e)
{
    var id = Guid.NewGuid().ToString().ToUpper();
    var name = Server.MapPath(string.Format("~/Temp/{0}.xls", id));
    file.Value = name;
}

```

```

e.File.SaveAs(name, true);

var book = new Workbook(name);

var data = book.Worksheets[0].Cells.ExportDataTable(0, 0,
book.Worksheets[0].Cells.Rows.Where(o => o[0].Value != null).Count(), 14, false);

grid.DataSource = data;

grid.DataBind();

}

```

12. 查询审计

//绑定图表系列

```

protected void BuildSeries(string title)
{
    ChartSeries series = new ChartSeries(title);
    this.c.Chart.Series.Add(series);
}

```

//绑定图表系列项

```

protected void BuildSeriesItem(ChartSeries series, double count, string label)
{
    ChartSeriesItem item = new ChartSeriesItem(count, label);
    item.Label.TextBlock.Appearance.TextProperties.Color = Color.HotPink;
    item.Label.TextBlock.Appearance.TextProperties.Font = new Font("Arial", 12, FontStyle.Bold);
    series.Items.Add(item);
}

```

//绑定图表数据源

```

protected void BindChartDataSource()

```

```

{
    this.UnDisplayTitle();

    this.ClearSeries();

    this.BuildSeries();

    List<string> l = new List<string>();

    foreach (var s in HomoryContext.Value.Department.Where(o => o.State == State.启用 &&
        o.Type == DepartmentType.学校).OrderBy(o => o.Ordinal).Select(o => o.Name).ToList())
    {
        if (CurrentRights.Contains(HomoryCoreConstant.RightGlobal))
        {
            l.Add(s);
        }
        else
        {
            var campus = CurrentCampus.Name;
            if (s == campus)
            {
                l.Add(s);
            }
        }
    }
    if (l.Count() > 0)
    {
        List<Calced> lA = new List<Calced>();
        var width = 0;
        foreach (var s in l)

```

```

{
IA.Add(new Calced(s, HomoryContext.Value.ViewQueryStudent.Count(o => o.学校 == s &&
o.状态 == "启用")));

width++;
}

foreach (Calced item in IA)
{
this.BuildSeriesItem(this.c.Series[SeriesIndexes[0]], (double)item.Num, item.Num.ToString());
}

this.c.Width = 150 + 50 * width;

this.c.PlotArea.XAxis.AutoScale = false;

this.c.PlotArea.XAxis.Clear();

this.c.PlotArea.XAxis.AddRange(1, l.Count, 1);

for (int i = 0; i < l.Count; i++)
{
this.c.PlotArea.XAxis[i].TextBlock.Appearance.AutoTextWrap = AutoTextWrap.True;

this.c.PlotArea.XAxis[i].TextBlock.Appearance.TextProperties.Color = (i % 2) == 0 ?
Color.Silver : Color.YellowGreen;

this.c.PlotArea.XAxis[i].TextBlock.Text = l.ElementAt(i);
}
}
}

//初始化筛选项

protected void InitFilterMenu(GridFilterMenu menu)

```

```

{
    menu.Items.Clear();

    var itemNoFilter = new RadMenuItem { Text = "全部", Value = "NoFilter" };
    menu.Items.Add(itemNoFilter);

    var itemContains = new RadMenuItem { Text = "模糊查找", Value = "Contains" };
    menu.Items.Add(itemContains);

    var itemEqualTo = new RadMenuItem { Text = "精确查找", Value = "EqualTo" };
    menu.Items.Add(itemEqualTo);
}

```

13. 个人设定

//上传个人头像

```

protected void upload_FileUploaded(object sender, Telerik.Web.UI.FileUploadedEventArgs e)
{
    var xName = CurrentUser.Id.ToString().ToUpper();

    if (CurrentUser.Teacher != null)
        xName = CurrentUser.Teacher.IDCard;

    if (CurrentUser.Student != null)
        xName = CurrentUser.Student.IDCard;

    var file = string.Format("~/Common/头像/用户/{0}.jpg", xName);

    e.File.SaveAs(Server.MapPath(file), true);

    viewer.ImageUrl = string.Format("{0}?{1}", file, Guid.NewGuid().ToString("N"));

    var id = CurrentUser.Id;

    var u = HomoryContext.Value.User.First(o => o.Id == id);

    u.Icon = file;
}

```

```

HomoryContext.Value.SaveChanges();

LogOp(OperationType.编辑);

}

//修改个人密码

protected void buttonSave_OnClick(object sender, EventArgs e)

{

    string k, v;

    var id = CurrentUser.Id;

    var u = HomoryContext.Value.User.First(o => o.Id == id);

    var length = int.Parse(HomoryContext.Value.ApplicationPolicy.Single(o => o.Name ==
    "UserPasswordLength" && o.ApplicationId == Guid.Empty).Value);

    if (string.IsNullOrEmpty(userPassword.Value) ||
    string.IsNullOrEmpty(userPassword2.Value) ||
    !userPassword.Value.Equals(userPassword2.Value, StringComparison.Ordinal) ||
    userPassword.Value.Length < length)

    {

        Notify(panel, string.Format("请输入不少于{0}位的密码，并确保两次输入的密码一致", length),
        "warn");

        return;

    }

    u.Password = HomoryCrytor.Encrypt(userPassword.Value, out k, out v);

    HomoryContext.Value.SaveChanges();

    LogOp(OperationType.编辑);

    try { UserHelper.UpdateUserPassword(id.ToString().ToUpper(), userPassword.Value); } catch {
    Notify(panel, "办公平台密码更新失败", "warn"); }

    viewer.ImageUrl = CurrentUser.Icon;

```



```
Notify(panel, "保存成功", "success");  
}
```

乐翼云资源平台

1. 基础方法

//互动操作日志

```
public static void ST_Resource(this Entities db, Guid id, ResourceOperationType  
operationType, decimal score = 0M)  
  
    {  
  
        var today = DateTime.Today;  
  
        var year = today.Year;  
  
        var month = today.Month;  
  
        var count = db.ResourceStatisticsMonthly.Count(o => o.Year == year && o.Month  
== month && o.Id == id);  
  
        if (count == 0)  
  
        {  
  
            db.ResourceStatisticsMonthly.Add(new ResourceStatisticsMonthly { Id = id, Year =  
year, Month = month, TimeStamp = DateTime.Now });  
  
            db.SaveChanges();  
  
        }  
  
        var ops = db.ResourceStatisticsMonthly.Single(o => o.Year == year && o.Month ==  
month && o.Id == id);  
  
        switch (operationType)  
  
        {  
  
            case ResourceOperationType.Comment:  
  
                ops.Comment++;  
  
            }  
  
        }  
  
    }
```

```

        break;

    case ResourceOperationType.Download:

        ops.Download++;

        break;

    case ResourceOperationType.Favourite:

        ops.Favourite++;

        break;

    case ResourceOperationType.Grade:

        ops.Grade += score;

        break;

    case ResourceOperationType.Rate:

        ops.Rate++;

        break;

    case ResourceOperationType.View:

        ops.View++;

        break;

    }

}

```

//互动操作统计

```

public static void LogOp(this Entities db, Guid userId, Guid campusId, ResourceLogType
logType, int? value = null)

{

    var today = DateTime.Today;

    var year = today.Year;

```

```
var month = today.Month;

var count = db.ResourceLog.Count(o => o.Year == year && o.Month == month && o.Id ==
userId);

if (count == 0)
{
    db.ResourceLog.Add(new ResourceLog
    {
        Id = userId,
        Year = year,
        Month = month,
        Article = 0,
        Courseware = 0,
        Paper = 0,
        Media = 0,
        View = 0,
        Favourite = 0,
        Download = 0,
        Comment = 0,
        Reply = 0,
        Rate = 0,
        Credit = 0,
        TimeStamp = DateTime.Now,
        CampusId = campusId
    });
}
```

```
db.SaveChanges();

}

var ops = db.ResourceLog.Single(o => o.Year == year && o.Month == month && o.Id ==
userId);

switch (logType)
{
case ResourceLogType.下载资源:
ops.Download++;

break;

case ResourceLogType.个人积分:
ops.Credit += value ?? 0;

break;

case ResourceLogType.发布文章:
ops.Article++;

break;

case ResourceLogType.发布视频:
ops.Media++;

break;

case ResourceLogType.发布试卷:
ops.Paper++;

break;

case ResourceLogType.发布课件:
ops.Courseware++;

break;
```

```

case ResourceLogType.回复评论:
ops.Reply++;
break;

case ResourceLogType.收藏资源:
ops.Favourite++;
break;

case ResourceLogType.浏览资源:
ops.View++;
break;

case ResourceLogType.评定资源:
ops.Rate++;
break;

case ResourceLogType.评论资源:
ops.Comment++;
break;
}

db.SaveChanges();
}

//标题太长时截取

public static string CutString(this string text, int? maxLength, string suffix = "")
{
    if (string.IsNullOrEmpty(text))
        return string.Empty;
    if (!maxLength.HasValue)

```

```

        maxLength = text.Length - 1;

return text.Length > maxLength

    ? string.Format("{0}{1}", text.Substring(0, maxLength.Value), suffix)

    : text;

}

```

//格式化日期时间

```

public static string FormatTime(this DateTime time)

{

    if (time.Date == DateTime.Today)

    {

        return time.ToString("HH:mm");

    }

    if ((DateTime.Today - time).TotalDays < 1)

        return "昨天";

    return (DateTime.Today - time).TotalDays < 2 ? "前天" : time.ToString("yyyy-MM-dd");

}

```

//格式化短格式日期时间

```

public static string FormatTimeShort(this DateTime time)

{

    if (time.Date == DateTime.Today)

    {

        return time.ToString("HH:mm");

    }

    if ((DateTime.Today - time).TotalDays < 1)

```

```

        return "昨天";

        return (DateTime.Today - time).TotalDays < 2 ? "前天" : time.ToString("MM/dd");
    }

    //根据标识获取资源
    protected Resource R(object id)
    {
        var gid = Guid.Parse(id.ToString());
        return HomoryContext.Value.Resource.Single(o => o.Id == gid);
    }

    //根据标识获取用户
    protected User U(object id)
    {
        var gid = Guid.Parse(id.ToString());
        return HomoryContext.Value.User.Single(o => o.Id == gid);
    }

    //根据标识获取用户所在的学校
    protected string UC(object id)
    {
        var gid = Guid.Parse(id.ToString());
        var user = HomoryContext.Value.User.Single(o => o.Id == gid);
        return user.DepartmentUser.Count(o => o.Type == DepartmentUserType.部门主职教师 ||
        o.Type == DepartmentUserType.借调后部门主职教师) > 0 ? "[" + user.DepartmentUser.First(o
        => o.Type == DepartmentUserType.部门主职教师 || o.Type == DepartmentUserType.借调后部
        门主职教师).TopDepartment.Name + "]" : "";
    }
}

```

2. 资源互动

//添加收藏

```
protected void addFavourite_OnClick(object sender, ImageClickEventArgs e)
{
    var id = Guid.Parse(((ImageButton)sender).CommandArgument);

    HomoryContext.Value.UserFavourite.AddOrUpdate(new UserFavourite { UserId =
    CurrentUser.Id, FavouriteUserId = id, State = State.启用 });

    HomoryContext.Value.SaveChanges();

    var favouritesSource = CurrentUser.MeFavourite.Where(o => o.State == State.启用).ToList();

    var favouritesSourceId = favouritesSource.Select(o => o.FavouriteUserId).ToList();

    favourites.DataSource = favouritesSource.Select(o => o.FavouriteUser).ToList();

    favourites.DataBind();

    relatives.DataSource =

    HomoryContext.Value.User.Where(o => o.State < State.审核).ToList().Where(o =>
    !favouritesSourceId.Contains(o.Id)).ToList();

    relatives.DataBind();
}
```

3. 资源板块

//绑定文章类资源

```
homory_article.DataSource = HomoryContext.Value.Resource.Where(o => o.Type ==
ResourceType.文章 && o.State == State.启用)

    .OrderByDescending(o => o.Time)

    .Take(Count)

    .ToList();

homory_article.DataBind();
```



```
//绑定课件类资源
```

```
public void BindCourseware()
{
    var repeaters = new Repeater[]
    {A0, A1, A2, A3, A4, A5, A6, B0, B1, B2, B3, B4, B5, B6, C0, C1, C2, C3, C4, C5, C6};

    foreach (var repeater in repeaters)
    {
        Func<Resource, bool> funcWhere;

        Func<Resource, dynamic> funcOrder;

        var index = int.Parse(repeater.ID.Substring(1, 1));

        var order = repeater.ID.Substring(0, 1);

        switch (order)
        {
            case "A":

                funcOrder = o => o.Time;

                break;

            case "B":

                funcOrder = o => o.View;

                break;

            default:

                funcOrder = o => o.Credit;

                break;

        }

        if (index < 6)
```

```

        {

funcWhere =

o => o.Type == ResourceType.课件 && o.State < State.审核 && o.CourseId != null &&
o.CourseId == Courses[index];

repeater.DataSource = HomoryContext.Value.Resource.Where(predicate:
funcWhere).OrderByDescending(funcOrder).Take(10).ToList();

repeater.DataBind();

}

else

{

funcWhere = o => o.Type == ResourceType.课件 && o.State < State.审核 && o.CourseId !=
null;

var courseP = Courses[index];

var coursesP = HomoryContext.Value.Catalog.Where(o => o.State < State.审核 && o.Type
== CatalogType.课程 && o.ParentId == courseP).ToList();

var source = coursesP.Join(HomoryContext.Value.Resource.Where(funcWhere), c => c.Id, r =>
r.CourseId, (c, r) => r).ToList();

repeater.DataSource = source;

repeater.DataBind();

}

}

//绑定名师工作室数据

studio.DataSource = HomoryContext.Value.Group.Where(o => o.State < State.审核 &&
o.Type == GroupType.名师团队).Take(5).ToList();

studio.DataBind();

public void BindVideo()

```

```

{
    video_latest.DataSource =
        HomoryContext.Value.Resource.Where(o => o.State < State.审核 && o.Type ==
ResourceType.视频)
            .OrderByDescending(o => o.Time)
            .Take(4)
            .ToList();
    video_latest.DataBind();
    video_popular.DataSource =
        HomoryContext.Value.Resource.Where(o => o.State < State.审核 && o.Type ==
ResourceType.视频)
            .OrderByDescending(o => o.View)
            .Take(4)
            .ToList();
    video_popular.DataBind();
}

```

```

var list = new List<Resource>();
if (IsOnline)
{
    var __courses = CurrentUser.Taught.Where(o => o.State < State.审核).Select(o =>
o.CourseId).ToList();
    foreach (var __c in __courses)
    {

```

```
list.AddRange(HomoryContext.Value.Resource.Where(o => o.State == State.启用 &&
o.UserId == CurrentUser.Id && o.ResourceCatalog.Count(p => p.CatalogId == __c &&
p.State < State.启用) > 0).Take(12).ToList());
```

```
if (list.Count >= 12)
```

```
break;
```

```
}
```

```
}
```

```
list = list.OrderByDescending(o=>o.Time).ToList();
```

```
if (list.Count < 12)
```

```
{
```

```
list.AddRange(HomoryContext.Value.Resource.Where(o => o.State < State.审核 && o.Type
== ResourceType.视频)
```

```
.OrderByDescending(o => o.Credit)
```

```
.Take(12 - list.Count)
```

```
.ToList());
```

```
}
```

```
video_random.DataSource = list.Take(4);
```

```
video_random.DataBind();
```

```
video_randomX.DataSource = list.Count > 4 ? list.Skip(4).Take(list.Count - 4) : list.Take(4);
```

```
video_randomX.DataBind();
```

```

video_cut.DataSource = HomoryContext.Value.ResourceComment.Where(o => o.State <
State.审核 && o.Timed == true).OrderByDescending(o => o.Time).Select(o =>
o.Resource).Distinct().Take(4).ToList();

video_cut.DataBind();

}

```

4. 资源发布

//创建用户个人资源文件夹

```

protected void CreateDirectories()

{

    var path = string.Format("../Common/资源/{0}/附件",
CurrentUser.Id.ToString().ToUpper());

    var dir = Server.MapPath(path);

    if (!Directory.Exists(dir))

        Directory.CreateDirectory(dir);

    path = string.Format("../Common/资源/{0}/文章", CurrentUser.Id.ToString().ToUpper());

    dir = Server.MapPath(path);

    if (!Directory.Exists(dir))

        Directory.CreateDirectory(dir);

    path = string.Format("../Common/资源/{0}/课件", CurrentUser.Id.ToString().ToUpper());

    dir = Server.MapPath(path);

    if (!Directory.Exists(dir))

        Directory.CreateDirectory(dir);

    path = string.Format("../Common/资源/{0}/试卷", CurrentUser.Id.ToString().ToUpper());

    dir = Server.MapPath(path);

    if (!Directory.Exists(dir))

```

```

        Directory.CreateDirectory(dir);

        path = string.Format("../Common/资源/{0}/视频", CurrentUser.Id.ToString().ToUpper());

        dir = Server.MapPath(path);

        if (!Directory.Exists(dir))

            Directory.CreateDirectory(dir);

        path = string.Format("../Common/资源/{0}/上传", CurrentUser.Id.ToString().ToUpper());

        dir = Server.MapPath(path);

        if (!Directory.Exists(dir))

            Directory.CreateDirectory(dir);
    }

```

//获取当前用户发布的资源类型

```

protected ResourceType ResourceType
{
    get
    {
        switch (Request.QueryString["Type"])
        {
            case "Courseware":
                return ResourceType.课件;

            case "Paper":
                return ResourceType.试卷;

            case "Media":
                return ResourceType.视频;

            case "Article":

```

```

        return ResourceType.文章;

    default:

        return ResourceType.文章;

    }

}

}

```

//添加资源标签

```

protected void publish_tag_add_OnClick(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(publish_tag_content.Text.Trim()))
        return;

    var rt = new ResourceTag
    {
        Tag = publish_tag_content.Text.Trim(),
        ResourceId = CurrentResource.Id,
        State = State.启用
    };

    HomoryContext.Value.ResourceTag.AddOrUpdate(rt);
    HomoryContext.Value.SaveChanges();
    publish_tag_content.Text = string.Empty;
    publish_tag_tags.DataSource =

        HomoryContext.Value.ResourceTag.Where(o => o.ResourceId ==
CurrentResource.Id && o.State == State.启用).Select(o => o.Tag).ToList();

    publish_tag_tags.DataBind();
}

```

```
}
```

```
//设置资源获奖情况
```

```
protected void publish_prize_range_OnSelectedIndexChanged(object sender,  
RadComboBoxSelectedIndexChangedEventArgs e)
```

```
{
```

```
    if (e.Value == "0")
```

```
    {
```

```
        CurrentResource.PrizeRange = null;
```

```
        CurrentResource.PrizeLevel = null;
```

```
        publish_prize_level.SelectedIndex = 0;
```

```
    }
```

```
    else
```

```
    {
```

```
        var range = (ResourcePrizeRange)(int.Parse(e.Value));
```

```
        CurrentResource.PrizeRange = range;
```

```
    }
```

```
    HomoryContext.Value.SaveChanges();
```

```
}
```

```
//设置资源发布所属目录
```

```
protected void publish_catalog_OnEntryAdded(object sender, DropDownTreeEntryEventArgs  
e)
```

```
{
```

```
    CatalogChozen(e, State.启用);
```

```
}
```


//发布资源

```
protected void publish_publish_go_OnClick(object sender, ImageClickEventArgs e)

{
    if (string.IsNullOrEmpty(publish_title_content.Value))
    {
publish_publish_panel.ResponseScripts.Add("popNotify()");

        return;
    }

    var resource = CurrentResource;

    var catalog = HomoryContext.Value.ResourceCatalog.Where(o => o.ResourceId ==
resource.Id && o.Catalog.Type == CatalogType.文章 && o.State < State.删除).FutureCount();

    var catalogX = HomoryContext.Value.ResourceCatalog.Where(o => o.ResourceId ==
resource.Id && o.Catalog.Type == CatalogType.视频 && o.State < State.删除).FutureCount();
    var ccc = CurrentResource.CourseId.HasValue ? 1 : 0;
    var ggc = CurrentResource.GradeId.HasValue ? 1 : 0;

    switch (resource.Type)
    {
        case ResourceType.文章:
            {
                if (catalog.Value == 0 && (ccc + ggc) == 0)
                {
                    publish_publish_panel.ResponseScripts.Add("popNotify()");

                    return;
                }
            }
        }
    }
```

```

        if (catalog.Value == 0 && (ccc + ggc) == 1)
        {
            publish_publish_panel.ResponseScripts.Add("popNotify()");
            return;
        }
        break;
    }

case ResourceType.课件:
case ResourceType.试卷:
    {
        if ((ccc + ggc) < 2)
        {
            publish_publish_panel.ResponseScripts.Add("popNotify()");
            return;
        }
        break;
    }

case ResourceType.视频:
    {
        if (catalogX.Value == 0 || (ccc + ggc) < 2)
        {
            publish_publish_panel.ResponseScripts.Add("popNotify()");
            return;
        }
    }

```

```

        break;
    }
}

if (string.IsNullOrEmpty(resource.Content) &&
string.IsNullOrEmpty(resource.Preview))
{
    publish_publish_panel.ResponseScripts.Add("popNotify()");
    return;
}

if (resource.PrizeRange != null && resource.PrizeLevel != null)
{
    var credit =

        HomoryContext.Value.PrizeCredit.SingleOrDefault(o => o.PrizeRange ==
resource.PrizeRange && o.PrizeLevel == resource.PrizeLevel);

    if (credit != null)
    {
        resource.Credit = credit.Credit;
LogOp(ResourceLogType.个人积分, resource.Credit);
    }
}

resource.Title = publish_title_content.Value;
resource.Content = publish_editor.Content;
resource.Time = DateTime.Now;
resource.State = State.启用;

```

```
        HomoryContext.Value.SaveChanges();

        LogOp(TeacherOperationType);

        Response.Redirect(string.Format("../Go/{1}?Id={0}", resource.Id, resource.Type ==
        Homory.Model.ResourceType.视频 ? "ViewVideo" : "ViewPlain"), false);

    }
}
```