拿一个id

string ID = db.UsersRepository.Where(x => x.UserName == UserName && x.UserPassword == Password).ToList().First().UserID;

//下拉$("#EnterpriseName option:first").prop("selected", 'selected');回到原来状态

--WebConfig配置--

<connectionStrings>

<add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;Initial Catalog=aspnet-Library-20191014094651;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|\aspnet-Library-20191014094651.mdf" providerName="System.Data.SqlClient" />

<add name="TradeDataBase" connectionString="server=172.16.12.6;database=TradeMent;uid=sa;pwd=123456;Max Pool Size=512;" providerName="System.Data.SqlClient" />

</connectionStrings>

--模板--

@{

ViewBag.Title = "我的反馈";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

--模板--

<!DOCTYPE html>

<html lang="zh">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta charset="utf-8" />

<title>@ViewBag.Title</title>

<link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />

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

<link href="~/Css/bootstrap.css" rel="stylesheet" />

<link href="~/Css/bootstrap.min.css" rel="stylesheet" />

<script src="https://cdn.staticfile.org/jquery/1.10.2/jquery.min.js">

</script>

@\* @Scripts.Render("~/bundles/jquery")

@RenderSection("scripts", required: false)\*@

</head>

<body>

<div id="body">

@RenderSection("featured", required: false)

<section class="content-wrapper main-content clear-fix">

@RenderBody()

</section>

</div>

<footer>

<div class="content-wrapper">

@\*visible-xs-block visible-sm-block\*@

<nav class="navbar navbar-default navbar-fixed-bottom">

<div class="container-fluid">

<div class="row" style="color: #316ea7;">

<div class="col-xs-3 col-sm-3" style="text-align: center;">

<a href="~/FoveManage/ExhibitionIndex" style="cursor: pointer; text-decoration: none;">

<div style="width: 100%; height: 25px; line-height: 30px;"><span class="glyphicon glyphicon-home" aria-hidden="true"></span></div>

<div style="width: 100%;"><span>首页</span> </div>

</a>

</div>

<div class="col-xs-3 col-sm-3" style="text-align: center;">

<a href="~/FoveManage/BoothApplicationIndex" style="cursor: pointer; text-decoration: none;">

<div style="width: 100%; height: 25px; line-height: 30px;"><span class="glyphicon glyphicon-plus" aria-hidden="true"></span></div>

<div style="width: 100%;"><span>展位申请</span> </div>

</a>

</div>

<div class="col-xs-3 col-sm-3" style="text-align: center;">

<a href="~/FoveManage/AnnuallyFileIndex" style="cursor: pointer; text-decoration: none;">

<div style="width: 100%; height: 25px; line-height: 30px;"><span class="glyphicon glyphicon-file" aria-hidden="true"></span></div>

<div style="width: 100%;"><span>年审材料</span> </div>

</a>

</div>

<div class="col-xs-3 col-sm-3" style="text-align: center;">

<a href="~/FoveManage/UserCenterIndex" style="cursor: pointer; text-decoration: none;">

<div style="width: 100%; height: 25px; line-height: 30px;"><span class="glyphicon glyphicon-user" aria-hidden="true"></span></div>

<div style="width: 100%;"><span>我的</span> </div>

</a>

</div>

</div>

</div>

</nav>

</div>

</footer>

<script src="~/Scripts/bootstrap.js"></script>

<script src="~/Scripts/bootstrap.min.js"></script>

</body>

</html>

--顶部导航条（两边都有图标，如果只有一个加height: 1px;就行了）--

<div class="container-fluid visible-xs-block visible-sm-block">

<div class="row" style="background: #41ace4; color: #fff; text-align: center; height: 40px; line-height: 40px;">

<div style="float: left; padding-left: 15px;">

<a href="/BillCenter/BillIndex" style="color: #fff;">

<span class="glyphicon glyphicon-menu-left" aria-hidden="true"></span>

</a>

</div>

<div style="font-size: 18px; display: inline;">编辑账单</div>

<div style="float: right; padding-right: 15px;" id="Btn\_Sure">

<span class="glyphicon glyphicon-ok" aria-hidden="true"></span>

</div>

</div>

</div>

--页面背景颜色覆盖--

<div style="position: absolute; top: 0; left: 0; width: 100%; min-height: 100%; height: auto; background: #F4F5F6;">

</div>

--Library.Repository--

--Constaint--

public partial class DataContent : DbContext

{

public DataContent()

: base("TradeDataBase")

{

}

public DbSet<Enterprise> EnterpriseRepository { get; set; }

public DbSet<EnterpriseAudit> EnterpriseAuditRepository { get; set; }

}

--Library.Repository--

--Entities--

[Table("Enterprise")]

public class Enterprise

{

[Key]

public string EnterpriseID { get; set; }

public string EnterpriseName { get; set; }

public string EnterpriseState { get; set; }

public Nullable<System.DateTime> AuthenticationTime { get; set; }

}

--Library.Repository--

--Enums--

/// <summary>

/// 用户身份

/// </summary>

public enum UsersRole

{

[Display(Name = "企业用户")]

EnterpriseUser = 0,

[Display(Name = "系统管理员")]

Administrator = 1,

[Display(Name = "出认证处管理员")]

ExitAdministrator = 2,

[Display(Name = "国际展览部管理员")]

InterAdministrator = 3

}

/// <summary>

/// 使馆认证状态

/// </summary>

public enum EmbassyRole

{

[Display(Name = "文件已受理")]

EmbassyGet = 1,

[Display(Name = "送外办")]

ForeignState = 2,

[Display(Name = "使馆认证完成")]

Completed = 3,

}

public class GetRole

{

//根据 名称 获取 Name

public static string GetEnumDesc(Enum en)//封装的函数，被底下的方法调用

{

Type type = en.GetType();

MemberInfo[] memInfo = type.GetMember(en.ToString());

if (memInfo != null && memInfo.Length > 0)

{

object[] attrs = memInfo[0].GetCustomAttributes(typeof(System.ComponentModel.DataAnnotations.DisplayAttribute), false);

if (attrs != null && attrs.Length > 0) return ((System.ComponentModel.DataAnnotations.DisplayAttribute)attrs[0]).Name;

}

return en.ToString();

}

//根据 名称 获取 数字

public static int GetEnumNum(Enum en)

{

var name = en.ToString();

int num = Convert.ToInt32(Enum.Parse(typeof(AnnualRole), name));

return num;

}

//根据 数字 获取 Name

public static string GetEnumName(int num)

{

AnnualRole name = (AnnualRole)Enum.ToObject(typeof(AnnualRole), num);

var en = GetEnumDesc(name);

return en;

}

}

GetRole.GetUserName(0);//调用

--Library.Service--

--BaseRepository--

public class BaseRepository

{

protected DataContent db = new DataContent();

}

--Library.Service--

--interface --

public interface IAnnuallyFileMent

{

AnnuallyFileDM AnnuallyFileList(String ID);//年审材料列表

String AnnuallyFileAdd(String UserID, ForeAnnuallyFileDM ForeAnnuallyFile);//年审材料的上传（添加）

AnnuallyFileDetailDM AnnuallyFileDetail(String ID);//年审材料详情

}

--Library.Service--

--method--

public class AnnuallyFileMent : BaseRepository, IAnnuallyFileMent

--模态框--

<input type="submit" value="添加用户" class="btn float-right bg-info" style="margin-right: 30px;font-size:14px;" data-toggle='modal' data-target='#myModalUserAdd'>

<form onsubmit="return false" id="formUserAdd" method="post">

<div class="modal fade" id="myModalUserAdd" tabindex="-1" role="dialog" aria-labelledby="myModalLabelUserAdd">

<div class="modal-dialog" role="document">

<div class="modal-content" style="background-color: #fff;">

<div class="modal-header">

<h4 class="modal-title" id="ModalBookAddLabel">添加用户</h4>

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

</div>

<div class="modal-body">

<div class="row" style="margin-bottom: 10px;">

<div class="col-md-4 col-xs-4" style="padding: 0px 10px 0 53px;">用户名</div>

<div class="col-md-6 col-xs-4" style="margin-bottom: 10px;">

<input type="text" name="UserName" placeholder="请输入用户名" class="form-control form-control-sm" id="UserNameAdd"/>

</div>

<div class="col-md-4" style="padding: 0px 10px 0 53px;">密码</div>

<div class="col-md-6 " style="margin-bottom: 10px;">

<input type="text" name="UserPassword" placeholder="请输入密码" class="form-control form-control-sm" id="UserPasswordAdd"/>

</div>

<div class="col-md-4" style="padding: 0px 10px 0 53px;">联系方式</div>

<div class="col-md-6 " style="margin-bottom: 10px;">

<input type="text" name="ContactPhone" placeholder="请输入电话号码" class="form-control form-control-sm" id="ContactPhoneAdd"/>

</div>

<div class="col-md-4" style="padding: 0px 10px 0 53px;">地址</div>

<div class="col-md-6 " style="margin-bottom: 10px;">

<input type="text" name="Address" placeholder="请输入地址" class="form-control form-control-sm" id="AddressAdd"/>

</div>

<div class="col-md-4" style="padding: 0px 10px 0 53px;">邮箱</div>

<div class="col-md-6 " style="margin-bottom: 10px;">

<input type="text" name="Mailbox" placeholder="请输入邮箱" class="form-control form-control-sm" id="MailboxAdd"/>

</div>

<div class="col-md-4" style="padding: 0px 10px 0 53px;">用户权限</div>

<div class="col-md-6 " style="margin-bottom: 10px;">

<div>

<label style="font-weight: inherit;">

<input type="radio" name="UserRole" id="Sex\_Man1" value="2" checked="checked">

出认证处管理员

</label>

<label style="font-weight: inherit;">

<input type="radio" name="UserRole" id="Sex\_WoMan1" value="3">

国际展览部管理员

</label>

<label style="font-weight: inherit;">

<input type="radio" name="UserRole" id="Sex\_WoMan2" value="0">

企业用户

</label>

</div>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default btn-sm" data-dismiss="modal" style="text-shadow: none;">关闭</button>

<button type="button" class="btn btn-primary btn-sm" id="BtnAdd">保存</button>

</div>

</div>

</div>

</div>

</form>

$("#btn\_EditGet").click(function () {

if (listid == undefined || listid == "" || listid == null) {

alert("请选中一行");

return false;

}

UserEditGet(listid);

});

--Session--

public class UserInfo

{

public static String UserID

{

get

{

object obj = HttpContext.Current.Session["\_USERID"];

return obj == null ? string.Empty : obj.ToString();

}

set

{

HttpContext.Current.Session["\_USERID"] = value;

}

}

public static String UserRole

{

get

{

object obj = HttpContext.Current.Session["\_USERROLE"];

return obj == null ? string.Empty : obj.ToString();

}

set

{

HttpContext.Current.Session["\_USERROLE"] = value;

}

}

public static String UserName

{

get

{

object obj = HttpContext.Current.Session["\_UserName"];

return obj == null ? string.Empty : obj.ToString();

}

set

{

HttpContext.Current.Session["\_UserName"] = value;

}

}

}

--中文搜索uri--

function getUrlVars() {  
            var vars = [], hash;  
            var hashes = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&');  
            for (var i = 0; i < hashes.length; i++) {  
                hash = hashes[i].split('=');  
                vars.push(hash[0]);  
                vars[hash[0]] = hash[1];  
            }  
            return vars;  
        }

 var params = getUrlVars();  
            localid = decodeURI(params.Text);

--页面懒加载--

<div id="test" style="margin-top:0px;">

</div>

<div id="pda" class="container-fluid">

<div class="row" style="text-align: center">

<div class="col-md-12 col-xs-12">

@\* <div id="pagedown" style="color:#999999";>加载更多..</div>\*@

<input type="text" name="page" id="pagenow" hidden="hidden" value="1" />

</div>

</div>

</div>

$(function () {

IndexAjax();

$(window).scroll(function () {

var scrollTop = $(this).scrollTop();

var scrollHeight = $(document).height();

var windowHeight = $(this).height();

if (scrollTop + windowHeight == scrollHeight) {

if (pd == true) {

fenye();

}

}

});

})

function fenye() {

var num = $('#pagenow').val();

num++;

if (num \* 10 >= Number(CountData)) {

$('#pda').hide();

pd = false;

}

$('#pagenow').val(num);

IndexAjax();

}

public ActionResult EmbassyMentList(int? page = 1)//首页列表（展会列表）

{

ITradeExhibitionMent FoveMethod = new TradeExhibitionMent();

int page1 = page.Value \* 10;

return Json(FoveMethod.EmbassyMentList(page1), JsonRequestBehavior.AllowGet);

}

public ExhibitionIndexDM EmbassyMentList(int? page = 1)//首页列表（展会列表）(可以拓展搜索做查询)

{

int SumNumber = page.Value;//每页几条数据

ExhibitionIndexDM model1 = new ExhibitionIndexDM();

var Exhibitions = (from m in db.ExhibitionRepository select m).ToList();

int CountPage = Exhibitions.Count();//总数据条数

//int AllPage = CountPage % SumNumber == 0 ? CountPage / SumNumber : CountPage / SumNumber + 1;//总页数

Exhibitions = Exhibitions.OrderByDescending(x => x.ExhibitionTime).Take(SumNumber).ToList();

model1.ExhibitionList = Exhibitions.ToList();

//model1.NowPage = page.Value;

//model1.AllPage = AllPage;

model1.CountPage = CountPage;

return model1;

}

--分页（改进版）--

function IndexAjax() {

$.ajax({

url: '@Url.Action("TradeFairApplyList", "TradeFairApply")',

type: "GET",

data: $('#formUserList').serialize(),

dataType: "json",

success: function (data, stutas, xhr) {

console.log(data);

var str1 = "";

var str2 = "";

$("#test").html("");

$("#paging").html("");

//$('#fy').empty();

$.each(data.ExhibitionApplyList, function (i, val) {

var date = ChangeDateFormat(val.ExhibitionTime);

str1 += "<tr id=" + val.ExhibitionID + ">"

+ "<td>"

+ "<a href=../TradeBackMent/TradeFairApplyDetail?ID=" + val.ExhibitionID + " >"

+ "<font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.ExhibitionTitle

+ "</font></font></a></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.ExhibitionPlace

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ date

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.CountApplicant

+ "</font></font></td>"

+ "</tr>"

});

$("#test").append(str1);

if (Number(data.CountPage) != 0) {

str2 += "<div class='jsgrid-pager'><font style='vertical-align: inherit;'>"

+ "<font style='vertical-align: inherit;'>页数：</font>"

var allpage = data.AllPage;//总页数

var nowpage = data.NowPage;//当前页

var uppage = Number(nowpage) - Number(1);//上一页

var downpage = Number(nowpage) + Number(1);//下一页

var time = 0;//循环执行次数

var bigtime;//最多显示几个页;

if (allpage > 5) {

bigtime = 5

}

else {

bigtime = allpage

}

var know = allpage - bigtime - nowpage + 1//我也不知道是啥，但是有用

console.log("know", know);

if (nowpage > 1) {

str2 += "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + 1 + ') " >'

+ "<font style='vertical-align: inherit;'>首页</font></a></span>"

+ "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(uppage) + ') " >'

+ "<font style='vertical-align: inherit;'>上一页</font></a></span>";

}

if (know >= 0) {

for (var i = nowpage; i > 0 && i <= Number(allpage) && time <= bigtime - 1 ; i++) {

time++;

if (i == nowpage) {

str2 += "<span class='jsgrid-pager-page jsgrid-pager-current-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;color:#212529;'>"

+ i

+ "</font></a></span>";

}

else {

str2 += "<span class='jsgrid-pager-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ i

+ "</font></a></span>";

}

}

}

else {

for (var i = allpage - bigtime + 1; i > 0 && i <= Number(allpage) && time <= bigtime - 1 ; i++) {

time++;

if (i == nowpage) {

str2 += "<span class='jsgrid-pager-page jsgrid-pager-current-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;color:#212529;'>"

+ i

+ "</font></a></span>";

}

else {

str2 += "<span class='jsgrid-pager-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ i

+ "</font></a></span>";

}

}

}

if (nowpage < allpage) {

str2 += "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(downpage) + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ "下一页"

+ "</font></a></span>"

+ "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(allpage) + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ "末页"

+ "</font></a></span>";

}

console.log("data.AllPage", data.AllPage);

str2 += "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ nowpage

+ "</font>"

+ "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ "of"

+ "</font>"

+ "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ allpage

+ "</font>"

+ "</div>";

}

else {

str2 += "<div style=margin-top:0px;>" + "暂无数据！" + "</div>";

}

$("#paging").append(str2);

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

function js\_method(page) {//分页事件

$('#page').val(page);

IndexAjax();

}

<div class="jsgrid-pager-container" id="paging" style="margin-left: 18px">

</div>

<input type="text" id="page" value="1" name="page" hidden="hidden" />

--身份过滤器--

[MyRoleFilter(Message = 2)]

在Filters文件夹里

public class MyRoleFilter:AuthorizeAttribute

{

public int Message { get; set; }

public override void OnAuthorization(AuthorizationContext filterContext)

{

if (!string.IsNullOrEmpty(UserInfo.UserRole))

{

//base.OnAuthorization(filterContext);

int session = Convert.ToInt32(UserInfo.UserRole);

if (session != Message)

{

UserInfo.UserRole = null;

//需要跳转到另一个页面用Result

if (session == 0)

{

filterContext.Result = new RedirectResult("/FoveManage/UserLogin");

}

else

{

filterContext.Result = new RedirectResult("/BackManage/TradeBackMent/AdminerLogin");

}

return;

}

}

else

{

UserInfo.UserRole= null;

filterContext.Result = new RedirectResult("/BackManage/TradeBackMent/AdminerLogin");

return;

}

}

}

--登录--

function BtnLogin() {

$.ajax({

url: '@Url.Action("Login", "TradeBackMent")',

type: "GET",

data: $('#formLogin').serialize(),

success: function (data, stutas, xhr) {

//$(this).attr('src', "../Util/ValidateCode.aspx?" + '+Math.random()');

if (data == null||data=="") {

alert("不存在该身份！");

reutrn;

}

if (data == "0") {

alert("抱歉，您没有该权限！");

//window.location.href = "/FoveManage/ExhibitionIndex";

}

else if (data == "1") {

//alert("登录成功！");

window.location.href = "/BackManage/TradeBackMent/UserMentIndex";

}

else if (data == "2") {

//alert("登录成功！");

window.location.href = "/BackManage/TradeBackMent/EnterpriseApplyIndex";

}

else if (data == "3") {

//alert("登录成功！");

window.location.href = "/BackManage/TradeBackMent/TradeFairMessageIndex";

}

else {

alert(data);

$('#imgValidateCode1').trigger("click");

$("[name='LoginCode']").val("");

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

$(function () {

$('#BtnLogin').click(function () {

BtnLogin();

})

});

//登录

public ActionResult Login(String UserName, String Password, String LoginCode)

{

IBackmentService FoveMethod = new BackmentService();

string ConfirmCode = Session["CheckCode"].ToString().ToLower();

if (FoveMethod.Login(UserName, Password, LoginCode, ConfirmCode) == "登录成功！")

{

string role = FoveMethod.LoginRole(UserName, Password).UserRole;

string id = FoveMethod.LoginRole(UserName, Password).UserID;

UserInfo.UserName = UserName;

UserInfo.UserID = id;

if (role == GetRole.GetEnumDesc(UsersRole.EnterpriseUser))

{

return Content("0");//这里因为是用户，所以不给他session存值（两个登录有点麻烦）

}

else if (role == GetRole.GetEnumDesc(UsersRole.Administrator))

{

UserInfo.UserRole = "1";

}

else if (role == GetRole.GetEnumDesc(UsersRole.ExitAdministrator))

{

UserInfo.UserRole = "2";

}

else if (role == GetRole.GetEnumDesc(UsersRole.InterAdministrator))

{

UserInfo.UserRole = "3";

}

else

{

UserInfo.UserRole =null;

}

return Content(UserInfo.UserRole);

}

return Content(FoveMethod.Login(UserName, Password, LoginCode, ConfirmCode));

}

public String Login(String UserName, String Password, String LoginCode, String ConfirmCode)//登录

{

if (string.IsNullOrEmpty(UserName))

{

return "用户名不能为空！";

}

if (string.IsNullOrEmpty(Password))

{

return "密码不能为空！";

}

if (string.IsNullOrEmpty(LoginCode))

{

return "验证码不能为空！";

}

if (ConfirmCode != LoginCode.ToLower())

{

return "请输入正确的验证码！";

}

var Verification = (from a1 in db.UsersRepository

where a1.UserID == (from a2 in db.UsersRepository where a2.UserName == UserName select a2.UserID).FirstOrDefault()

&& a1.UserPassword == Password

select a1).Count();

if (Verification > 0)

{

return "登录成功！";

//登录成功

}

else

{

return "用户名或账号错误！";

//登录失败

}

}

public AdminerRoleDM LoginRole(String UserName, String Password)//登录判断身份传入session

{

AdminerRoleDM model1 = new AdminerRoleDM();

string Role = db.UsersRepository.Where(x => x.UserName == UserName && x.UserPassword == Password).ToList().First().UserRole;

string ID = db.UsersRepository.Where(x => x.UserName == UserName && x.UserPassword == Password).ToList().First().UserID;

model1.UserRole = Role;

model1.UserID = ID;

return model1;

}

--分页（原始版本）--

<div id="fy">

</div>

<input type="text" id="page" value="1" name="page" hidden="hidden" />

<div>

<button type="button" style="border: 0; background: none" id="BtnFirst">首页</button>

<button type="button" style="border: 0; background: none" id="BtnUp">上一页</button>

<button type="button" style="border: 0; background: none" id="BtnDown">下一页</button>

<button type="button" style="border: 0; background: none" id="BtnLast">尾页</button>

</div>

var resultValue;//全局变量

$(function () {

IndexAjax();

$('#btn\_Search').click(function () {

IndexAjax();

})

$('#BtnFirst').click(function () {

$('#page').val("1");

IndexAjax();

})

$('#BtnUp').click(function () {

$('#page').val(resultValue.NowPage <= 1 ? resultValue.NowPage : resultValue.NowPage - 1);

IndexAjax();

})

$('#BtnDown').click(function () {

$('#page').val(resultValue.NowPage >= resultValue.AllPage ? resultValue.NowPage : resultValue.NowPage + 1);

IndexAjax();

})

$('#BtnLast').click(function () {

$('#page').val(resultValue.AllPage);

IndexAjax();

})

});

function IndexAjax() {

$.ajax({

url: '@Url.Action("IndexList", "BookMent")',

type: "GET",

data: $('#formBookList').serialize(),

dataType: "json",

success: function (data, stutas, xhr) {

resultValue = data;

console.log(data);

var str1 = "";

//清空table中的html

$("#test").html("");

//循环语句输出list数据

$.each(data.bookList, function (i, val) {

var date = ChangeDateFormat(val.AddDate);

var number = i + 1;

str1 = "<tr id=" + val.BookID + ">"

//+ "<td>" + "<div style='display:none'>" + "<input type='text' id='firstname' value=" + val.BookID + ">" + "</div>" + "</td>"

+"<td>"+number+"</td>"

+"<td>"+val.BookName+"</td>"

+"<td>"+val.Author+"</td>"

+"<td>"+val.Class+"</td>"

+"<td>"+val.Publish+"</td>"

+"<td>"+val.Price+"</td>"

+"<td>"+date+"</td>"

+ "<td>" + "<button type='button' name='delete' class='btn btn-default' value=" + val.BookID + ">删除" + "</button>" + "</td>"

+"</tr>"

$("#test").append(str1);

});

$('#test').on("click", "tr", function () {

$(this).parent().find("tr.focus").toggleClass("focus");

$(this).toggleClass("focus");

listid = $(this).attr("id");

console.log(listid);

})

$("#Btn\_Detail").click(function () {

if (listid == undefined || listid == "" || listid == null) {

alert("请选中一行");

return false;

}

DetailAjax(listid);

});

$($("[name='delete']")).bind("click", function () {

console.log($(this).val());

if (confirm("确认删除吗?")) {

//DeleteAjax($(this).val());

var id = $(this).val();

//胜利ar link = $(this);

$.post("@Url.Action("Delete","BookMent")", { "id": id }, function (data) {

alert(data);

IndexAjax();

});

}

});

//分页

$('#fy').empty();

if (Number(data.CountPage) != 0) {

$('#fy').append("当前第：" + data.NowPage + "页 ，总共：" + data.AllPage + "页");

console.log(Number(data.CountPage));

}

else {

$('#fy').append("<div style=margin-top:60px;>" + "暂无数据！" + "</div>");

console.log(Number(data.CountPage));

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

--导出--

<a href="@Url.Action("FileExport", "TradeFairMessage")" >

<div class="btn float-right bg-info" style="margin-right: 30px;">导出数据</div>

</a>

//导出数据

public ActionResult FileExport()

{

ITradeFairMessage TradeFairMethod = new TradeFairMessage();

//TradeFairMethod.FileExportList();//执行接口

//FileExportDM model1 = new FileExportDM();

var Result = TradeFairMethod.FileExportList().ExhibitionList;

//var Result = model1.ExhibitionList;

HSSFWorkbook ExcelBook = new HSSFWorkbook();

NPOI.SS.UserModel.ISheet sheet1 = ExcelBook.CreateSheet("展会信息");

// 第一列

NPOI.SS.UserModel.IRow row = sheet1.CreateRow(0);

row.CreateCell(0).SetCellValue("商展标题");

row.CreateCell(1).SetCellValue("企业名称");

row.CreateCell(2).SetCellValue("展览地点");

row.CreateCell(3).SetCellValue("展览简介");

row.CreateCell(4).SetCellValue("联系人");

row.CreateCell(5).SetCellValue("传真");

row.CreateCell(6).SetCellValue("联系电话");

row.CreateCell(7).SetCellValue("展览内容");

row.CreateCell(8).SetCellValue("展览时间");

//IDataFormat dataformat = ExcelBook.CreateDataFormat();

//ICellStyle style1 = ExcelBook.CreateCellStyle();

//style1.DataFormat = dataformat.GetFormat("yyyy年MM月dd日 hh:mm");

for (int i = 0; i < Result.Count(); i++)

{

NPOI.SS.UserModel.IRow rowtemp = sheet1.CreateRow(i + 1);

rowtemp.CreateCell(0).SetCellValue(Result[i].ExhibitionTitle);

rowtemp.CreateCell(1).SetCellValue(Result[i].EnterpriseName);

rowtemp.CreateCell(2).SetCellValue(Result[i].ExhibitionPlace);

rowtemp.CreateCell(3).SetCellValue(Result[i].ExhibitionSynopsis);

rowtemp.CreateCell(4).SetCellValue(Result[i].Contacts);

rowtemp.CreateCell(5).SetCellValue(Result[i].Fax);

rowtemp.CreateCell(6).SetCellValue(Result[i].ContactPhone);

rowtemp.CreateCell(7).SetCellValue(Result[i].ExhibitionContent);

rowtemp.CreateCell(8).SetCellValue(Result[i].ExhibitionTime.ToString());

}

var fileName = "展会表.xls";

MemoryStream ExcelStream = new MemoryStream();

//string urlPath = "/Execl/" + fileName; // 文件下载的URL地址，供给前台下载

//string filePath = System.Web.HttpContext.Current.Server.MapPath("\\" + urlPath); // 文件路径

//FileStream file = new FileStream(filePath, FileMode.Create);

//ExcelBook.Write(file);

ExcelBook.Write(ExcelStream);

ExcelStream.Seek(0, SeekOrigin.Begin);

return File(ExcelStream, "application/vnd.ms-excel", fileName); //return Content(TF.CreateExcel());

}

public FileExportDM FileExportList()//导出数据

{

FileExportDM model1 = new FileExportDM();

var Exhibitions = (from m in db.ExhibitionRepository select m).ToList();

int CountPage = Exhibitions.Count();//总数据条数

Exhibitions = Exhibitions.OrderByDescending(x => x.ExhibitionTime).ToList();

model1.ExhibitionList = Exhibitions.ToList();

model1.CountPage = CountPage;

return model1;

}

public class FileExportDM

{

public int CountPage;

//public IEnumerable<Exhibition> ExhibitionList

//{

// get;

// set;

//}

public List<Exhibition> ExhibitionList { get; set; }

}

--列表选中一行--

<script>

var listid;

function IndexAjax() {

$.ajax({

url: '@Url.Action("UserMentIndexList", "UserMent")',

type: "GET",

data: $('#formUserList').serialize(),

dataType: "json",

success: function (data, stutas, xhr) {

console.log(data);

var str1 = "";

var str2 = "";

$("#test").html("");

$("#paging").html("");

//$('#fy').empty();

$.each(data.UsersList, function (i, val) {

str1 += "<tr id=" + val.UserID + ">"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.UserName

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.UserRole

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.ContactPhone

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.Address

+ "</font></font></td>"

+ "<td><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.Mailbox

+ "</font></font></td>"

+ "<td>" + "<button type='button' name='delete' class='btn\_operation' value=" + val.UserID + " style='color: transparent;text-shadow: 0 0 0 #000;'>删除" + "</button>" + "</td>"

+ "</tr>"

});

$("#test").append(str1);

$('#test').on("click", "tr", function () {

$(this).parent().find("tr.focus").toggleClass("focus");

$(this).toggleClass("focus");

listid = $(this).attr("id");

console.log(listid);

})

$("#btn\_EditGet").click(function () {

if (listid == undefined || listid == "" || listid == null) {

alert("请选中一行");

return false;

}

UserEditGet(listid);

});

$($("[name='delete']")).bind("click", function () {

console.log($(this).val());

if (confirm("确认删除吗?")) {

var id = $(this).val();

$.post("@Url.Action("Delete","UserMent")", { "UserId": id }, function (data) {

alert(data);

IndexAjax();

});

}

});

if (Number(data.CountPage) != 0) {

str2 += "<div class='jsgrid-pager'><font style='vertical-align: inherit;'>"

+ "<font style='vertical-align: inherit;'>页数：</font>"

var allpage = data.AllPage;//总页数

var nowpage = data.NowPage;//当前页

var uppage = Number(nowpage) - Number(1);//上一页

var downpage = Number(nowpage) + Number(1);//下一页

var time = 0;//循环执行次数

var bigtime;//最多显示几个页;

if (allpage > 5) {

bigtime = 5

}

else {

bigtime = allpage

}

var know = allpage - bigtime - nowpage + 1//我也不知道是啥，但是有用

console.log("know", know);

if (nowpage > 1) {

str2 += "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + 1 + ') " >'

+ "<font style='vertical-align: inherit;'>首页</font></a></span>"

+ "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(uppage) + ') " >'

+ "<font style='vertical-align: inherit;'>上一页</font></a></span>";

}

if (know >= 0) {

for (var i = nowpage; i > 0 && i <= Number(allpage) && time <= bigtime - 1 ; i++) {

time++;

if (i == nowpage) {

str2 += "<span class='jsgrid-pager-page jsgrid-pager-current-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;color:#212529;'>"

+ i

+ "</font></a></span>";

}

else {

str2 += "<span class='jsgrid-pager-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ i

+ "</font></a></span>";

}

}

}

else {

for (var i = allpage - bigtime + 1; i > 0 && i <= Number(allpage) && time <= bigtime - 1 ; i++) {

time++;

if (i == nowpage) {

str2 += "<span class='jsgrid-pager-page jsgrid-pager-current-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;color:#212529;'>"

+ i

+ "</font></a></span>";

}

else {

str2 += "<span class='jsgrid-pager-page'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + i + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ i

+ "</font></a></span>";

}

}

}

if (nowpage < allpage) {

str2 += "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(downpage) + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ "下一页"

+ "</font></a></span>"

+ "<span class='jsgrid-pager-nav-button'>"

+ '<a href="javascript:void(0);" Onclick="js\_method(' + Number(allpage) + ') " >'

+ "<font style='vertical-align: inherit;'>"

+ "末页"

+ "</font></a></span>";

}

console.log("data.AllPage", data.AllPage);

str2 += "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ nowpage

+ "</font>"

+ "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ "of"

+ "</font>"

+ "<font style='vertical-align: inherit;padding: 0px 3px;'>"

+ allpage

+ "</font>"

+ "</div>";

}

else {

str2 += "<div style=margin-top:0px;>" + "暂无数据！" + "</div>";

}

$("#paging").append(str2);

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

function js\_method(page) {//分页事件

$('#page').val(page);

IndexAjax();

}

function BtnAdd() {

$.ajax({

url: '@Url.Action("UserAdd", "UserMent")',

type: "post",

data: $('#formUserAdd').serialize(),

success: function (data, stutas, xhr) {

alert(data);

if (data === "用户添加成功！") {

$('#myModalUserAdd').modal('hide');

IndexAjax();

$("#UserNameAdd").val("");

$("#UserPasswordAdd").val("");

$("#ContactPhoneAdd").val("");

$("#AddressAdd").val("");

$("#MailboxAdd").val("");

//$("#Sex\_Man1").attr('checked', false);

//$("#Sex\_WoMan1").attr('checked', false);

//$("#Sex\_WoMan2").attr('checked', false);

$("#Sex\_Man1").prop("checked", true);

$("#Sex\_WoMan1").prop("checked", false);

$("#Sex\_WoMan2").prop("checked", false);

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

function UserEditGet(id) {

$.ajax({

url: '@Url.Action("UserEditGet", "UserMent")',

type: "get",

data: { "UserId": id },

success: function (data, stutas, xhr) {

console.log(data);

if (data != null) {

$('#UserId').val(data.UserID);

$('#UserNameEdit').val(data.UserName);

$('#UserPasswordEdit').val(data.UserPassword);

$("#ContactPhoneEdit").val(data.ContactPhone);

$("#AddressEdit").val(data.Address);

$("#MailboxEdit").val(data.Mailbox);

if (data.UserRole == "出认证处管理员") {

$("#UserRole\_1").prop("checked", true);

$("#UserRole\_2").prop("checked", false);

$("#UserRole\_3").prop("checked", false);

}

if (data.UserRole == "国际展览部管理员") {

$("#UserRole\_1").prop("checked", false);

$("#UserRole\_2").prop("checked", true);

$("#UserRole\_3").prop("checked", false);

}

if (data.UserRole == "企业用户") {

$("#UserRole\_1").prop("checked", false);

$("#UserRole\_2").prop("checked", false);

$("#UserRole\_3").prop("checked", true);

}

}

else {

alert("发生错误！");

return false;

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

function BtnSave() {

$.ajax({

url: '@Url.Action("UserEditPost", "UserMent")',

type: "post",

data: $('#formUserEdit').serialize(),

success: function (data, stutas, xhr) {

alert(data);

if (data === "修改成功！") {

$('#myModalUserEdit').modal('hide');

IndexAjax();

}

else {

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

$(function () {

IndexAjax();

$('#btn\_Search').click(function () {

IndexAjax();

})

$('#BtnAdd').click(function () {

BtnAdd();

})

$('#BtnEdit').click(function () {

BtnSave();

})

});

</script>

tr.focus {

background: #e4e4e4;

}

--弹窗--

return Content("<script>alert('修改成功！');window.location.href='/BackMananger/EmbassyAccreditation/EmbassyAccreditationIndex';</script>");

--导入--

[https://www.cnblogs.com/pdbs/p/9474748.html](qq://txfile/)

--编辑账单--

$('#img1').attr("src", data.BillKindImg);

$("#" + data.BillKindID).addClass("img\_border");//向选中的img添加边框

--获取时间--

function ChangeDateFormat(jsondate) {

jsondate = jsondate.replace("/Date(", "").replace(")/", "");

if (jsondate.indexOf("+") > 0) {

jsondate = jsondate.substring(0, jsondate.indexOf("+"));

}

else if (jsondate.indexOf("-") > 0) {

jsondate = jsondate.substring(0, jsondate.indexOf("-"));

}

var date = new Date(parseInt(jsondate, 10));

var month = date.getMonth() + 1 < 10 ? "0" + (date.getMonth() + 1) : date.getMonth() + 1;

var currentDate = date.getDate() < 10 ? "0" + date.getDate() : date.getDate();

return date.getFullYear() + "-" + month + "-" + currentDate;

};

var date = ChangeDateFormat(data.ExhibitionTime);

--获取uri--

function UrlSearch() {

var name, value;

var str = location.href; //获取到整个地址

var num = str.indexOf("?")

str = str.substr(num + 1); //取得num+1后所有参数，这里的num+1是下标 str.substr(start [, length ]

var arr = str.split("&"); //以&分割各个参数放到数组里

for (var i = 0; i < arr.length; i++) {

num = arr[i].indexOf("=");

if (num > 0) {

name = arr[i].substring(0, num);

value = arr[i].substr(num + 1);

this[name] = value;

return value;

}

}

}

$(function () {

localid = UrlSearch();

IndexAjax(localid);

$("#test").on("click", function () {

PageNav\_BottomShow();

});

});

--时间分组--

function IndexAjax() {

$.ajax({

url: '@Url.Action("MyMessageList", "UserCenter")',

type: "GET",

dataType: "json",

success: function (data, stutas, xhr) {

console.log(data);

var str1 = "";

$("#test").html("");

if (Number(data.CountPage) > 0) {

var arr = data.LeaveMessageList;

for (var x = 0; x < arr.length; x++) {//向arr对象数组添加AddTime字段，因为接下来需要用时间分组

arr[x]['AddTime'] = arr[x].LeaveMessageTime;

}

$.each(arr, function (i, val) {//将日期字段转换成年月日格式，接下来用来分组

arr[i].LeaveMessageTime = ChangeDateFormat(val.LeaveMessageTime);

});

console.log("arr", arr);

var map = {},//用时间分组arr数组

dest = [];

for (var i = 0; i < arr.length; i++) {

//console.log(i);

var ai = arr[i];

if (!map[ai.LeaveMessageTime]) {

dest.push({

LeaveMessageTime: ai.LeaveMessageTime,

data: [ai]

});

map[ai.LeaveMessageTime] = ai;

}

else {

for (var j = 0; j < dest.length; j++) {

var dj = dest[j];

if (dj.LeaveMessageTime == ai.LeaveMessageTime) {

dj.data.push(ai);

break;

}

}

}

}

var sj = JSON.stringify(dest);

console.log("sj",sj);

var day2 = new Date();//获取当天时间

day2.setTime(day2.getTime());

var s2 = day2.getFullYear() + "年" + (day2.getMonth() + 1) + "月" + day2.getDate() + "日";

console.log(s2);

console.log(dest.length);

for (var x = 0; x < dest.length; x++) {//遍历数组输出值

//console.log(dest[x].AddDate);

//console.log(dest[x].data);

str1 += "<div class='time-label'>";

if (dest[x].LeaveMessageTime == s2) {

str1 += "<span class='bg-red'><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>";

}

else {

str1 += "<span class='bg-green'><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>";

}

str1 += dest[x].LeaveMessageTime

+ "</font></font></span></div>";

$.each(dest[x].data, function (i, val) {//遍历数组输出值

console.log("sb1");

var hourtime = formatDate(val.AddTime);

str1 += "<div>"

+ "<i class='fas fa-comments bg-info'></i>"

+ "<div class='timeline-item'>"

+ "<span class='time'><i class='fas fa-clock'></i><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ hourtime

+ "</font></font></span>"

+ "<h3 class='timeline-header'>"

+"<font style='vertical-align: inherit;'>您对</font>"

+ "<a href=../FoveManage/ExhibitionDetail?ID=" + val.ExhibitionID + "><font style='vertical-align: inherit;'><font style='vertical-align: inherit;'>"

+ val.ExhibitionTitle

+ "</font></font></a><font style='vertical-align: inherit;'>发表了评论</font></h3>"

+ "<div class='timeline-body'><font style='vertical-align: inherit;'>"

+ val.LeaveMessageContent

+ "</font></div>"

+ "<div class='timeline-footer'>"

+ "</div></div></div>";

});

}

if (dest.length > 0) {

str1 += "<div><i class='fas fa-clock bg-gray'></i></div>";

$("#test").addClass("timeline");//添加样式

}

else {

str1 += "<div style=margin-top:0px;margin-left:5px;>" + "暂无数据！" + "</div>";

}

}

else {

str1 += "<div class='row' style='margin-top: 46%'>"

+ "<div class='col-sm-12 col-xs-12 phone\_padding'>"

+ "<img src='../Images/foveimg\_2.png' class='img-responsive center-block' width='186'/>"

+ "</div>"

+ "<div class='col-sm-12 col-xs-12 phone\_padding' style='margin-top: 15px;text-align:center;'>"

+ "您还没有发表过评论~"

+ "</div>"

+ "</div>";

}

$("#test").append(str1);

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

function ChangeDateFormat(jsondate) {

jsondate = jsondate.replace("/Date(", "").replace(")/", "");

if (jsondate.indexOf("+") > 0) {

jsondate = jsondate.substring(0, jsondate.indexOf("+"));

}

else if (jsondate.indexOf("-") > 0) {

jsondate = jsondate.substring(0, jsondate.indexOf("-"));

}

var date = new Date(parseInt(jsondate, 10));

var month = date.getMonth() + 1 < 10 ? "0" + (date.getMonth() + 1) : date.getMonth() + 1;

var currentDate = date.getDate() < 10 ? "0" + date.getDate() : date.getDate();

return date.getFullYear() + "年" + month + "月" + currentDate + "日";

};

function formatDate(val) {

if (val == undefined) {

return '';

}

var reg = /^\/Date\(\d+\)\/$/;

if (!reg.test(val)) return '';//格式不正确 ，返回空

var strDate = val.substr(1, val.length - 2);

var obj = eval('(' + "{ date :new " + strDate + "}" + ')')

var date = obj.date;

var year = date.getFullYear();

var month = date.getMonth() + 1 < 10 ? '0' + (date.getMonth() + 1) : date.getMonth() + 1;

var day = date.getDate() < 10 ? '0' + date.getDate() : date.getDate();

var datetime = "";

var hour = date.getHours() < 10 ? '0' + date.getHours() : date.getHours();

var minute = date.getMinutes() < 10 ? '0' + date.getMinutes() : date.getMinutes();

var seconds = date.getSeconds() < 10 ? '0' + date.getSeconds() : date.getSeconds();

return datetime + ' ' + hour + ':' + minute;

}

--列表--

//商展申请审核列表

public ActionResult TradeFairApplyList(String ExhibitionTitle, String ExhibitionPlace, String StartTime, String EndTime, int? page = 1)

{

ITradeFairApply TradeFairApplyMethod = new TradeFairApply();

return Json(TradeFairApplyMethod.TradeFairApplyList(ExhibitionTitle, ExhibitionPlace, StartTime, EndTime, page), JsonRequestBehavior.AllowGet);

}

public TradeFairApplyListDM TradeFairApplyList(String ExhibitionTitle, String ExhibitionPlace, String StartTime, String EndTime, int? page = 1)//商展申请审核的列表

{

var Exhibitions = (from a1 in db.ExhibitionRepository

//join a2 in db.ExhibitionApplyRepository on a1.ExhibitionID equals a2.ExhibitionID

select new ExhibitionList

{

ExhibitionID = a1.ExhibitionID,

ExhibitionTitle = a1.ExhibitionTitle,

ExhibitionPlace=a1.ExhibitionPlace,

ExhibitionSynopsis=a1.ExhibitionSynopsis,

Contacts=a1.Contacts,

Fax=a1.Fax,

ContactPhone=a1.ContactPhone,

ExhibitionContent=a1.ExhibitionContent,

ExhibitionTime=a1.ExhibitionTime,

CountApplicant = (from b in db.ExhibitionRepository

join c in db.ExhibitionApplyRepository on b.ExhibitionID equals c.ExhibitionID

where b.ExhibitionID == a1.ExhibitionID

select b.ExhibitionID).Count()//申请人数

//CountApplicant=a2.UserID.Count()

}).ToList();//寻找当前id的数据

int SumNumber = 10;//每页几条数据

TradeFairApplyListDM model1 = new TradeFairApplyListDM();

if (!String.IsNullOrEmpty(ExhibitionTitle))

{

Exhibitions = Exhibitions.Where(s => s.ExhibitionTitle.Contains(ExhibitionTitle)).ToList();

}

if (!String.IsNullOrEmpty(ExhibitionPlace))

{

Exhibitions = Exhibitions.Where(s => s.ExhibitionPlace.Contains(ExhibitionPlace)).ToList();

}

if (!String.IsNullOrEmpty(StartTime))

{

Exhibitions = Exhibitions.Where(s => s.ExhibitionTime >= Convert.ToDateTime(StartTime)).ToList();

}

if (!String.IsNullOrEmpty(EndTime))

{

Exhibitions = Exhibitions.Where(s => s.ExhibitionTime <= Convert.ToDateTime(EndTime)).ToList();

}

int CountPage = Exhibitions.Count();//总数据条数

int AllPage = CountPage % SumNumber == 0 ? CountPage / SumNumber : CountPage / SumNumber + 1;//总页数

Exhibitions = Exhibitions.OrderByDescending(x => x.ExhibitionTime).Skip((page.Value - 1) \* SumNumber).Take(SumNumber).ToList();

model1.ExhibitionApplyList = Exhibitions.ToList();

model1.NowPage = page.Value;

model1.AllPage = AllPage;

model1.CountPage = CountPage;

return model1;

}

public class TradeFairApplyListDM

{

public int CountPage;

public int NowPage;

public int AllPage;

public IEnumerable<ExhibitionList> ExhibitionApplyList

{

get;

set;

}

}

public class ExhibitionList

{

public string ExhibitionID { get; set; }

public string ExhibitionTitle { get; set; }

public string ExhibitionPlace { get; set; }

public string ExhibitionSynopsis { get; set; }

public string Contacts { get; set; }

public string Fax { get; set; }

public string ContactPhone { get; set; }

public string ExhibitionContent { get; set; }

public Nullable<System.DateTime> ExhibitionTime { get; set; }

public int CountApplicant { get; set; }//申请人数

}

--列表修改状态--

function ApplicantOperation(id, State) {//签证受理

if (confirm("确认签证受理吗?")) {

$.ajax({

url: '@Url.Action("ApplicantOperation","TradeFairApply")',

type: "POST",

data: { "EmbassyAplyId": id, "EmbassyAplyState": State },

success: function (data, stutas, xhr) {

if (data != null && data != "") {

alert(data);

IndexAjax(localid);

}

else {

alert("不存在该ID！");

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

}

+ "<button type='button' name='revert' class='btn\_operation' value=" + val.ExhibitionApplyID + " onclick='ApplicantOperation(this.value,5)' style='color: transparent;text-shadow: 0 0 0 #000;'>签证受理" + "</button>"

[HttpPost]

public ActionResult ApplicantOperation(String EmbassyAplyId, int EmbassyAplyState)

{

ITradeFairApply TradeFairApplyMethod = new TradeFairApply();

return Content(TradeFairApplyMethod.ApplicantOperation(EmbassyAplyId, EmbassyAplyState));

}

//申请人的列表操作(签证受理)

public String ApplicantOperation(String EmbassyAplyId, int EmbassyAplyState)

{

ExhibitionApply exhibitionApply = new ExhibitionApply();

exhibitionApply = db.ExhibitionApplyRepository.Find(EmbassyAplyId);

if (exhibitionApply == null)

{

return null;

}

string State = GetRole.GetCommercialName(EmbassyAplyState);

exhibitionApply.ExamineState = State;

Visa visa = new Visa();

visa.VisaID = DateTime.Now.ToString("yyyyMMddhhmmss");

visa.UserID = exhibitionApply.UserID;

visa.ExhibitionApplyID = EmbassyAplyId;

visa.VisaState = GetRole.GetVisaName(1);

visa.VisaTime = DateTime.Now;

db.VisaRepository.Add(visa);

db.SaveChanges();

//if (State == 2)

//{

// return "展会申请已通过！";

//}

//if (State == 3)

//{

// return "展会申请已驳回！";

//}

//if (State == 5)

//{

// return "展会申请审核完成！";

//}

return "签证受理完成！";

}

--清除button样式--

.btn\_operation {

border: 0px;

text-shadow: none;

background: 0px;

font-weight: 500;

padding-left: 0px;

-webkit-tap-highlight-color: rgba(0, 0, 0, 0);

-webkit-user-modify: read-write-plaintext-only;

outline: none;

--增删改查综合--

//用户管理

// GET: /BackManage/UserMent/

public ActionResult UserMentIndexList(String UserName, int UserRole, int? page = 1)//用户管理列表

{

IUserMent UserMethod = new UserMent();

return Json(UserMethod.UserMentList(UserName, UserRole, page), JsonRequestBehavior.AllowGet);

}

[HttpPost, ActionName("Delete")]

public ActionResult DeleteConfirmed(string UserId)

{

IUserMent UserMethod = new UserMent();

return Content(UserMethod.UserMentDelete(UserId));

}

//用户添加

[HttpPost]

public ActionResult UserAdd(String UserName, String UserPassword, String ContactPhone, String Address, String Mailbox, int UserRole)

{

IUserMent UserMethod = new UserMent();

return Content(UserMethod.UserMentAdd(UserName,UserPassword,ContactPhone,Address,Mailbox,UserRole));

}

//用户修改的Get方法

public ActionResult UserEditGet(String UserId)

{

IUserMent UserMethod = new UserMent();

if (UserMethod.UserMentEditGet(UserId) == null)

{

return Content(null);

}

return Json(UserMethod.UserMentEditGet(UserId), JsonRequestBehavior.AllowGet);

}

//用户修改的Post方法

[HttpPost]

public ActionResult UserEditPost(String UserId, String UserName, String UserPassword, String ContactPhone, String Address, String Mailbox, int UserRole)

{

IUserMent UserMethod = new UserMent();

return Content(UserMethod.UserMentEditPost(UserId,UserName,UserPassword,ContactPhone,Address,Mailbox,UserRole));

}

public UserMentListDM UserMentList(String UserName, int UserRole, int? page = 1)//用户管理的列表

{

int SumNumber = 10;//每页几条数据

UserMentListDM model1 = new UserMentListDM();

var Users = (from m in db.UsersRepository select m).ToList();

if (!String.IsNullOrEmpty(UserName))

{

Users = Users.Where(s => s.UserName.Contains(UserName)).ToList();

}

if (UserRole!=-1)

{

Users = Users.Where(s => s.UserRole.Equals(GetRole.GetUserName(UserRole))).ToList();

}

int CountPage = Users.Count();//总数据条数

int AllPage = CountPage % SumNumber == 0 ? CountPage / SumNumber : CountPage / SumNumber + 1;//总页数

Users = Users.OrderByDescending(x => x.CreationTime).Skip((page.Value - 1) \* SumNumber).Take(SumNumber).ToList();

model1.UsersList = Users.ToList();

model1.NowPage = page.Value;

model1.AllPage = AllPage;

model1.CountPage = CountPage;

return model1;

}

public String UserMentDelete(String UserId)//用户管理的删除

{

Users UserInfo = db.UsersRepository.Find(UserId);

if (UserInfo == null)

{

return "删除失败！";

}

var UserInfos = (from a1 in db.ExhibitionApplyRepository

where a1.UserID ==UserId

select a1

).Count();

if (UserInfos > 0)

{

return "删除失败，该用户已申请商展！";

}

db.UsersRepository.Remove(UserInfo);

db.SaveChanges();

return "删除成功！";

}

public String UserMentAdd(String UserName, String UserPassword, String ContactPhone, String Address, String Mailbox, int UserRole)//用户管理的添加

{

if (String.IsNullOrEmpty(UserName))

{

return "用户名不能为空！";

}

if (String.IsNullOrEmpty(UserPassword))

{

return "密码不能为空！";

}

if (String.IsNullOrEmpty(ContactPhone))

{

return "电话号码不能为空！";

}

if (String.IsNullOrEmpty(Address))

{

return "地址不能为空！";

}

if (String.IsNullOrEmpty(Mailbox))

{

return "邮箱不能为空！";

}

if (String.IsNullOrEmpty(UserRole.ToString()))

{

return "请选择用户权限！";

}

var UserInfos = (from a1 in db.UsersRepository

where a1.UserID == (from a2 in db.UsersRepository where a2.UserName == UserName select a2.UserID).FirstOrDefault()

select a1).Count();

if (UserInfos > 0)

{

return "用户名已存在！";

}

Users UserInfo = new Users();

UserInfo.UserID = DateTime.Now.ToString("yyyyMMddhhmmss");

UserInfo.UserName = UserName;

UserInfo.UserPassword = UserPassword;

UserInfo.ContactPhone = ContactPhone;

UserInfo.Address = Address;

UserInfo.Mailbox = Mailbox;

UserInfo.UserRole = GetRole.GetUserName(UserRole);

UserInfo.CreationTime = DateTime.Now;

db.UsersRepository.Add(UserInfo);

db.SaveChanges();

return "用户添加成功！";

}

public UserEditDM UserMentEditGet(String UserId)//用户管理的修改的Get方法

{

Users UserInfo = db.UsersRepository.Find(UserId);

UserEditDM model1 = new UserEditDM();

if (UserInfo == null)

{

return null;

}

model1.UserID = UserInfo.UserID;

model1.UserName = UserInfo.UserName;

model1.UserPassword = UserInfo.UserPassword;

model1.UserRole = UserInfo.UserRole;

model1.EnterpriseName = UserInfo.EnterpriseName;

model1.EnterpriseContacts = UserInfo.EnterpriseContacts;

model1.EnterpriseQQNumber = UserInfo.EnterpriseQQNumber;

model1.ContactPhone = UserInfo.ContactPhone;

model1.Address = UserInfo.Address;

model1.Mailbox = UserInfo.Mailbox;

model1.PostalCode = UserInfo.PostalCode;

model1.UserJurisdiction = UserInfo.UserJurisdiction;

model1.CreationTime = UserInfo.CreationTime;

return model1;

}

public String UserMentEditPost(String UserId, String UserName, String UserPassword, String ContactPhone, String Address, String Mailbox, int UserRole)//用户管理的修改的Post方法

{

if (String.IsNullOrEmpty(UserName))

{

return "用户名不能为空！";

}

if (String.IsNullOrEmpty(UserPassword))

{

return "密码不能为空！";

}

if (String.IsNullOrEmpty(ContactPhone))

{

return "电话号码不能为空！";

}

if (String.IsNullOrEmpty(Address))

{

return "地址不能为空！";

}

if (String.IsNullOrEmpty(Mailbox))

{

return "邮箱不能为空！";

}

if (String.IsNullOrEmpty(UserRole.ToString()))

{

return "请选择用户权限！";

}

var UserInfos = (from a1 in db.UsersRepository

where a1.UserID == (from a2 in db.UsersRepository where a2.UserName == UserName select a2.UserID).FirstOrDefault()

where a1.UserID != UserId

select a1).Count();

if (UserInfos > 0)

{

return "用户名已存在！";

}

Users UserInfo = new Users();

UserInfo = db.UsersRepository.Find(UserId);

if (UserInfo == null)

{

return null;

}

UserInfo.UserName = UserName;

UserInfo.UserPassword = UserPassword;

UserInfo.ContactPhone = ContactPhone;

UserInfo.Address = Address;

UserInfo.Mailbox = Mailbox;

UserInfo.UserRole = GetRole.GetUserName(UserRole);

db.SaveChanges();

return "修改成功！";

}

--表连接--

public AnnuallyFileDM AnnuallyFileList(String ID)//年审材料列表

{

if (string.IsNullOrEmpty(ID) || ID == null)//没有登录

{

return null;

}

////int SumNumber = 10;//每页几条数据

AnnuallyFileDM model1 = new AnnuallyFileDM();

var ExhibitionApplys = (from a1 in db.EnterpriseAuditRepository

join a2 in db.UsersRepository on a1.UserID equals a2.UserID

where a1.UserID == ID

select new PhoneEnterpriseAuditnList

{

EnterpriseName = a2.EnterpriseName,

EnterpriseAuditID = a1.EnterpriseAuditID,

UserID = a1.UserID,

PaymentVoucher = a1.PaymentVoucher,

BusinessLicense = a1.BusinessLicense,

ExamineState = a1.ExamineState,

ExamineTime = a1.ExamineTime

}).ToList();//寻找当前id的数据

int CountPage = ExhibitionApplys.Count();//总数据条数

//int AllPage = CountPage % SumNumber == 0 ? CountPage / SumNumber : CountPage / SumNumber + 1;//总页数

ExhibitionApplys = ExhibitionApplys.OrderByDescending(x => x.ExamineTime).ToList();

model1.EnterpriseAuditList = ExhibitionApplys.ToList();

//model1.NowPage = page.Value;

//model1.AllPage = AllPage;

model1.CountPage = CountPage;

return model1;

}

public class AnnuallyFileDM

{

public int CountPage;

public int NowPage;

public int AllPage;

public IEnumerable<PhoneEnterpriseAuditnList> EnterpriseAuditList

{

get;

set;

}

}

public class PhoneEnterpriseAuditnList

{

public string EnterpriseAuditID { get; set; }

public string UserID { get; set; }

public string PaymentVoucher { get; set; }

public string BusinessLicense { get; set; }

public string ExamineState { get; set; }

public Nullable<System.DateTime> ExamineTime { get; set; }

public string EnterpriseName { get; set; }

}

--js去重（下拉列表显示）--

function NavAjax() {

$.ajax({

url: '@Url.Action("EmbassyListNav", "EmbassyMent")',

type: "GET",

dataType: "json",

success: function (data, stutas, xhr) {

console.log(data);

var str1 = "";

//清空table中的html

$("#EnterpriseName").html("");

str1 += "<option value='请选择'>" + "请选择" + "</option>";

var arr = data.EnterpriseList;

for (var i = 0; i < arr.length; i++) {

for (var j = i + 1; j < arr.length; j++) {

if (arr[i].EnterpriseName == arr[j].EnterpriseName) { //第一个等同于第二个，splice方法删除第二个

arr.splice(j, 1);

j--;

}

}

}

//循环语句输出list数据

$.each(arr, function (i, val) {

str1 += "<option value=" + val.EnterpriseName + ">" + val.EnterpriseName + "</option>";

});

$("#EnterpriseName").append(str1);

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

--上传（一个）--

function BtnAdd() {

var fd = new FormData();

fd.append("ExhibitionImage", $('#file-0')[0].files[0]);

fd.append("EnterpriseName", $('#EnterpriseName').val());

fd.append("ExhibitionTitle", $('#ExhibitionTitle').val());

fd.append("ExhibitionPlace", $('#ExhibitionPlace').val());

fd.append("ExhibitionSynopsis", $('#ExhibitionSynopsis').val());

fd.append("Contacts", $('#Contacts').val());

fd.append("Fax", $('#Fax').val());

fd.append("ContactPhone", $('#ContactPhone').val());

fd.append("ExhibitionTime", $('#ExhibitionTime').val());

fd.append("ExhibitionContent", $('#ExhibitionContent').val());

$.ajax({

url: "@Url.Action("TradeFairMessageAdd", "TradeFairMessage")",//调用控制器中的方法

type: "post",

//dataType: "json",

data: fd,

async: false,

cache: false,

contentType: false,

processData: false,

success: function (data) {

alert(data);

if (data == "商展添加成功！") {

window.location.href = "../TradeBackMent/TradeFairMessageIndex";

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

<input id="file-0" name="file1" type="file" multiple="" class="file" data-overwrite-initial="false" data-min-file-count="1" data-max-file-count="1">

//商展信息添加

[HttpPost]

public ActionResult TradeFairMessageAdd(String ExhibitionImage, String EnterpriseName, String ExhibitionTitle, String ExhibitionPlace, String ExhibitionSynopsis, String Contacts, String Fax, String ContactPhone, String ExhibitionTime, String ExhibitionContent)//一个的

{

ITradeFairMessage TradeFairMethod = new TradeFairMessage();

string fileSave = Server.MapPath("~/upload/");

string path1 = "";

try

{

HttpFileCollectionBase file = Request.Files;

if (file.Count != 0)

{

for (int i = 0; i < file.Count; i++)

{

if (file.AllKeys[i] == "ExhibitionImage")

{

HttpPostedFileBase file1 = file[i];

string extName = Path.GetExtension(file1.FileName);

string newName = Guid.NewGuid().ToString() + extName;

file1.SaveAs(Path.Combine(fileSave, newName));

path1 = "/upload/" + newName;

}

}

}

}

catch

{

return Content("抱歉，您选择文件过大请重新选择");

}

return Content(TradeFairMethod.TradeFairMessageAdd(path1, EnterpriseName, ExhibitionTitle, ExhibitionPlace, ExhibitionSynopsis, Contacts, Fax, ContactPhone, ExhibitionTime, ExhibitionContent));

}

--上传（两个的）--

<input id="file-0" name="file1" type="file" multiple class="file" data-overwrite-initial="false" data-min-file-count="1" data-max-file-count="1" accept="image/\*">

<input id="file-1" name="file2" type="file" onchange="inputfile()" multiple class="file" data-overwrite-initial="false" data-min-file-count="2" data-max-file-count="2" accept="image/\*">//注意这里的onchange方法

var arr = new Array();

function inputfile() {

if (arr.length == 2) {

arr[1] = ($('#file-1')[0].files[1]);

}

else {

arr.push($('#file-1')[0].files[0]);

if ($('#file-1')[0].files[1] != null) {

arr.push($('#file-1')[0].files[1]);

}

}

}

function upload() {

var fd = new FormData();

fd.append("PayCard", $('#file-0')[0].files[0]);

for (var i = 0; i < arr.length; i++) {

fd.append("file", arr[i]);

}

$.ajax({

url: "@Url.Action("upload", "AnnuallyFile")",//调用控制器中的方法

type: "post",

//dataType: "json",

data: fd,

async: false,

cache: false,

contentType: false,

processData: false,

success: function (data) {

//alert(data);

$("#alertvalue").html("");

$("#alertvalue").append(data);

syalert.syopen('alert3');

if (data == "年审材料上传成功！") {

window.location.href = "../FoveManage/AnnuallyFileIndex";

}

},

error: function (jqXHR, textStatus, errorThrown) {

/\*弹出jqXHR对象的信息\*/

console.log(jqXHR.responseText, jqXHR.status, jqXHR.readyState, jqXHR.statusText, textStatus, errorThrown);

}

});

}

public String AnnuallyFileAdd(String UserID, ForeAnnuallyFileDM ForeAnnuallyFile)//年审材料的上传（添加）

{

//这里用的模型类存的值，不用应该也可以

if (String.IsNullOrEmpty(UserID))

{

return "您过久未操作，请重新登录！";

}

if (String.IsNullOrEmpty(ForeAnnuallyFile.PaymentVoucher))

{

return "付款凭证不能为空！";

}

if (String.IsNullOrEmpty(ForeAnnuallyFile.BusinessLicense))

{

return "营业执照不能为空！";

}

string[] arr = ForeAnnuallyFile.BusinessLicense.Split(',');

if (arr.Length < 2)

{

return "营业执照需要上传两张图片";

}

EnterpriseAudit Exhibition = new EnterpriseAudit();

Exhibition.EnterpriseAuditID = DateTime.Now.ToString("yyyyMMddhhmmss");

Exhibition.UserID = UserID;

Exhibition.PaymentVoucher = ForeAnnuallyFile.PaymentVoucher;

Exhibition.BusinessLicense = ForeAnnuallyFile.BusinessLicense;

Exhibition.ExamineState = GetRole.GetEnumName(1);

Exhibition.ExamineTime = DateTime.Now;

db.EnterpriseAuditRepository.Add(Exhibition);

db.SaveChanges();

return "年审材料上传成功！";

}

string[] arr = ForeAnnuallyFile.BusinessLicense.Split(',');

if (arr.Length < 2)

{

return "营业执照需要上传两张图片";

}

int count = 0;

[HttpPost]

public ActionResult upload(ForeAnnuallyFileDM ForeAnnuallyFile)

{

IAnnuallyFileMent AnnuallyFileMethod = new AnnuallyFileMent();

string fileSave = Server.MapPath("~/upload/");

try

{

HttpFileCollectionBase file = Request.Files;

if (file.Count != 0)

{

for (int i = 0; i < file.Count; i++)

{

HttpPostedFileBase file2 = file[i];

if (file.AllKeys[i] == "file")

{

count++;

string extName = Path.GetExtension(file2.FileName);

string newName = Guid.NewGuid().ToString() + extName;

file2.SaveAs(Path.Combine(fileSave, newName));

if (count == 1)

{

ForeAnnuallyFile.BusinessLicense = "/upload/" + newName;

}

else

{

ForeAnnuallyFile.BusinessLicense += "," + "/upload/" + newName;

}

}

if (file.AllKeys[i] == "PayCard")

{

string extName = Path.GetExtension(file2.FileName);

string newName = Guid.NewGuid().ToString() + extName;

file2.SaveAs(Path.Combine(fileSave, newName));

ForeAnnuallyFile.PaymentVoucher = "/upload/" + newName;

}

}

}

}

catch

{

return Content("抱歉，您选择文件过大请重新选择");

}

return Content(AnnuallyFileMethod.AnnuallyFileAdd(UserInfo.UserID, ForeAnnuallyFile));

}

--全选--

<script>

//点击全选，子复选框被选中

function demo() {

var allcheck = document.getElementById("allcheck");

var choice = document.getElementsByName("choice");

for (var i = 0; i < choice.length; i++) {

choice[i].checked = allcheck.checked;

}

}

//点击子复选框,全选框 选中、取消

function setAll() {

if (!$(".checknum").checked) {

$("#allcheck").prop("checked", false); // 子复选框某个不选择，全选也被取消

}

var choicelength = $("input[type='checkbox'][class='checknum']").length;

var choiceselect = $("input[type='checkbox'][class='checknum']:checked").length;

if (choicelength == choiceselect) {

$("#allcheck").prop("checked", true); // 子复选框全部部被选择，全选也被选择；1.对于HTML元素我们自己自定义的DOM属性，在处理时，使用attr方法；2.对于HTML元素本身就带有的固有属性，在处理时，使用prop方法。

}

}

</script>

<input type="checkbox" id="allcheck" onclick="demo()" class='gcs-checkbox'>

+ "<input type='checkbox' id=" + val.EnterpriseAuditID + " class='gcs-checkbox'onclick='setAll()' name='choice' value=" + val.EnterpriseAuditID + " >"

//disabled = 'disabled'禁用