**MVC+EF 快速搭建 基础信息维护功能**

1 Model配置

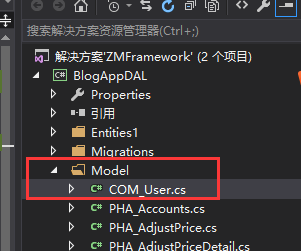
2 Controller开发

3 View开发

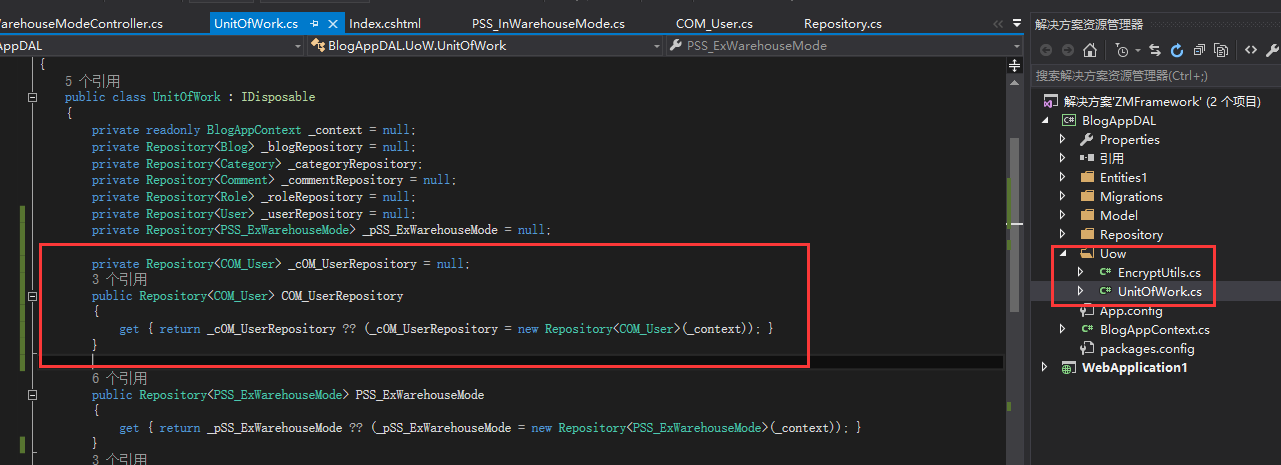
例如：用户信息维护如下

# 1 Model配置

## 1.1 新建用户实体

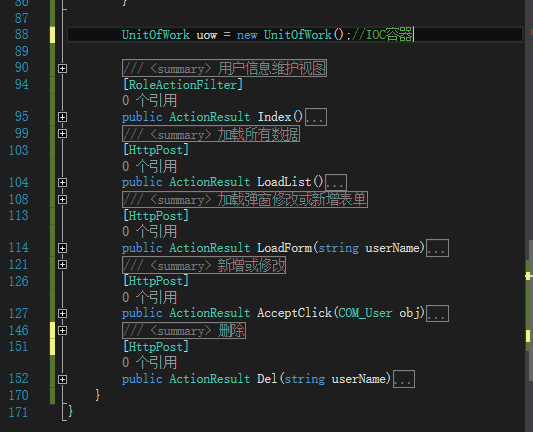


## 1.2 添加IOC 对应契约



# 2 Controller开发

## 2.1 概括

添加IOC容器，添加RoleActionFilter过滤器，添加LoadList()加载所有数据的方法，添加LoadForm(string UserName) 加载修改弹窗的初始数据，添加AcceptClick(obj)新增或修改的保存方法,添加Del(string userName)删除数据的方法2.2 

## 2.2 详细代码

### 2.2.1 Using引用

using System.Web.Mvc;//MVC库

using WebApplication1.Filter;//自定义过滤器

using BlogAppDAL;//数据实体

using BlogAppDAL.UoW;//IOC容器

### 2.2.2 RoleActionFilter过滤器，

using System.Web.Mvc;

namespace WebApplication1.Filter

{

/// <summary>

/// 登录过滤器

/// </summary>

public class RoleActionFilterAttribute:ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

base.OnActionExecuting(filterContext);

HttpContextBase context = filterContext.HttpContext;

HttpCookie cookie=context.Request.Cookies["user"];

if (cookie == null)

context.Response.Redirect("../User/login");

}

}

}

### 2.2.3 初始化IOC

UnitOfWork uow = new UnitOfWork();//IOC容器

### 2.2.4 控制器方法

#### 2.2.4.1 概括

1添加LoadList() 加载所有数据的方法，

2添加LoadForm(string UserName) 加载修改弹窗的初始数据，

3添加AcceptClick(obj) 新增或修改的保存方法,

4添加Del(string userName) 删除数据的方法

#### 2.2.4.2 详细代码

UnitOfWork uow = new UnitOfWork();//IOC容器

/// <summary>

/// 用户信息维护视图

/// </summary>

/// <returns></returns>

[RoleActionFilter]

public ActionResult Index()

{

return View();

}

/// <summary>

/// 加载所有数据

/// </summary>

/// <returns></returns>

[HttpPost]

public ActionResult LoadList()

{

return Json(uow.COM\_UserRepository.GetAll());

}

/// <summary>

/// 加载弹窗修改或新增表单

/// </summary>

/// <param name="userName"></param>

/// <returns></returns>

[HttpPost]

public ActionResult LoadForm(string userName)

{

COM\_User user = null;

if (!string.IsNullOrEmpty(userName))

user = uow.COM\_UserRepository.Get(b => b.Name == userName);

return Json(user);

}

/// <summary>

/// 新增或修改

/// </summary>

/// <param name="obj"></param>

/// <returns></returns>

[HttpPost]

public ActionResult AcceptClick(COM\_User obj)

{

int isOk = 0;

//判断userName的值 ，默认是 空，插入记录。非空代表数据库已经存在，则修改

isOk = uow.COM\_UserRepository.GetAll(b=>b.Name==obj.Name).ToList().Count;

try

{

if (isOk>0)

uow.COM\_UserRepository.Update(obj);

else

uow.COM\_UserRepository.Insert(obj);

uow.SaveChanges();

isOk = 1;//操作成功

}

catch (Exception)

{

isOk = 0;//操作失败

}

return Content(isOk.ToString());

}

/// <summary>

/// 删除

/// </summary>

/// <param name="userName"></param>

/// <returns></returns>

[HttpPost]

public ActionResult Del(string userName)

{

int isOk = 1;

var obj = uow.COM\_UserRepository.Get(b => b.Name == userName);

try

{

if (!string.IsNullOrEmpty(userName))

{

uow.COM\_UserRepository.Delete(obj);

uow.SaveChanges();

}

}

catch (Exception)

{

isOk = 0;

}

return Content(isOk.ToString());

}

# 3 View开发

## 3.1 概括

1. EasyUI基础框架引用
2. 页面DOM

2.1 datagrid

2.2 Form

1. 页面JS
   1. 页面初始化
   2. dataGrid 初始化
   3. Dialog初始化
   4. 表单验证和提交

## 3.2 详细代码

### 3.1EasyUI基础框架

@\*Easyui需要引入的文件\*@

<script type="text/javascript" src="~/Content/jquery-easyui-1.4.5/jquery.min.js"></script>

<script type="text/javascript" src="~/Content/jquery-easyui-1.4.5/jquery.easyui.min.js"></script>

<link href="~/Content/jquery-easyui-1.4.5/themes/default/easyui.css" rel="stylesheet" />

<link href="~/Content/jquery-easyui-1.4.5/themes/icon.css" rel="stylesheet" />

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

<script src="~/Scripts/messages\_zh.js"></script>

<script type="text/javascript" src="~/Content/Site.js"></script>

### 3.2页面DOM

#### 3.2.1 datagrid

<div>

<input id="del\_id" type="hidden" value=""/>

<div style="position:fixed;width:100%;height:100%">

<table id="grid"></table>

</div>

<div id="tb" style="padding:3px">

<a id="add" href="#" class="easyui-linkbutton" data-options="plain:true,iconCls:'icon-add'">新增</a>

<a id="del" href="#" class="easyui-linkbutton" data-options="plain:true,iconCls:'icon-remove'">删除</a>

<a id="btnReload" href="#" class="easyui-linkbutton" data-options="plain:true,iconCls:'icon-reload'">刷新</a>

</div>

</div>

#### 3.2.2 Form

<div id="dd" class="easyui-dialog">

<form id="form" name="form" method="post">

<table style="margin:8px">

<tr>

<td>编号:</td>

<td><input type="text" id="Code" name="Code" value="" disabled="disabled" /></td>

</tr>

<tr>

<td>账号:</td>

<td><input type="text" id="Username" name="Username" value="" class="required" /></td>

</tr>

<tr>

<td>密码:</td>

<td><input type="text" id="Password" name="Password" value="" class="required" /></td>

</tr>

<tr>

<td>姓名:</td>

<td><input type="text" id="Name" name="Name" value="" class="required" /></td>

</tr>

<tr>

<td>角色:</td>

<td><input type="text" id="Role" name="Role" value="" class="required" /></td>

</tr>

<tr>

<td>电话:</td>

<td><input type="text" id="Tel" name="Tel" value="" class="required" /></td>

</tr>

<tr>

<td>地址:</td>

<td><input type="text" id="Address" name="Address" value="" class="required" /></td>

</tr>

<tr>

<td>部门:</td>

<td><input type="text" id="Department" name="Department" value="" class="required" /></td>

</tr>

</table>

<div id="dlg-buttons">

<a id="ok" href="#" class="easyui-linkbutton" iconcls="icon-ok">确定</a>

<a id="cancel" href="#" class="easyui-linkbutton" iconcls="icon-cancel">取消</a>

</div>

</form>

</div>

### 3.3页面JS

#### 3.3.1 页面初始化

$(function () {

//初始化datagrid

InitGrid();

InitDialog();

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

$("#grid").datagrid("reload");

});

//ajax提交修改表单

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

$('#form').submit();

});

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

$('#dd').dialog('close');

});

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

$('#form')[0].reset();

$('#Code').val("此字段自动生成。");

$('#dd').dialog('open');

});

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

getAjax("../User/Del", { Username: $('#del\_id').val() }, function (data) {

if ($('#del\_id').val() == null || $('#del\_id').val() == "")

return false;

if (data == '1') {

$.messager.alert('提示', '操作失成功!');

$('#dd').dialog('close');

$("#grid").datagrid("reload");

return true;

}

else {

$.messager.alert('提示', '操作失败!');

return false;

}

});

});

});

#### 3.3.2 dataGrid 初始化

//实现对DataGird控件的绑定操作

function InitGrid() {

$('#grid').datagrid({

url: '../User/LoadList',

fit: true,

nowrap: true,//单行显示

autoRowHeight: false,

striped: false, //斑马纹

singleSelect: true,

collapsible: true, //可折叠

pagination: true,

pageSize: 20,

fitColumns: true, //自适应列宽

rownumbers: true,

columns: [[

{ title: '编号', field: 'Code', hidden: true },

{ title: '账号', field: 'Username', width: 10 },

{ title: '密码', field: 'Password', width: 10 },

{ title: '姓名', field: 'Name', width:10 },

{ title: '角色', field: 'Role', width: 10 },

{ title: '电话', field: 'Tel', width: 10 },

{ title: '地址', field: 'Address', width: 10 },

{ title: '部门', field: 'Department', width: 10 },

]],

toolbar: '#tb',

onDblClickRow: function (rowIndex, rowData) {

getAjax("../User/LoadForm",

{ Username: rowData['Username'] }, function (data) {

var data = eval("(" + data + ")");

SetWebControls(data);

});

$('#dd').dialog('open');

},

onClickRow: function (index, row) {

$('#del\_id').val(row['Username']);

}

})

var p = $('#grid').datagrid('getPager');

$(p).pagination({

beforePageText: '第',

afterPageText: '页 共 {pages} 页',

displayMsg: '当前显示 {from} - {to} 条记录 共 {total} 条记录',

});

};

#### 3.3.3 Dialog初始化

//实现对弹窗表单的数据绑定

function InitDialog() {

$('#dd').dialog({

title: '修改信息',

width: 400,

height: 400,

top: ($(window).height() - 400) \* 0.5, //居中

left: ($(window).width() - 400) \* 0.5,

closed: true,

cache: false,

modal: true,

buttons: '#dlg-buttons'

});

}

#### 3.3.4 表单验证和提交

<script>

//表单验证和提交

$(function () {

$('#form').validate({

rules: {

Name: {

required: true

}

},

submitHandler: function (form) {

var postData = GetWebControls("#form");

getAjax("/User/AcceptClick",

postData, function (data) {

if (data=='1') {

$.messager.alert('提示', '操作成功!');

$('#dd').dialog('close');

$("#grid").datagrid("reload");

return true;

}

else {

$.messager.alert('提示', '操作失败!');

return false;

}

});

},

invalidHandler: function (form, validator) { //不通过回调

return false;

},

showErrors: function (errorMap, errorList) {

this.defaultShowErrors();

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

$(errorList[i].element).one("blur", function () {

$("label.error[for='" + (this.id ? this.id : this.name) + "']").remove();

});

}

}

});

});

</script>