Bootstrap 开发框架-界面代码分析

版本: 3.0 编制人: 伍华聪

序号	修改人	修改日期	修改后版本	修改说明
1	伍华聪	2015-12-4	V1.0	正式版
2	伍华聪	2017-07-23	V2.0	完善修订版
3	伍华聪	2017-9-27	V3.0	完善并整合最新插件使用

日	퐢
н	

1.	引言.	5	;
1.1.	背景	5	5
1.2.	编写目	的	5
1.3.	参考资	料	5
1.4.	术语缩	写及约定	5
2.	框架界	界面代码的分析	5
2.1.	视图界	面代码介绍	5
2.2.	框架布	局设置8	3
2.3.	分页查	询12	2
	2.3.1.	控制器后台的分页支持15	5
	2.3.2.	界面的条件内容17	7
	2.3.3.	界面的分页代码19)
	2.3.4.	在代码生成工具中集成分页支持22	2
2.4.	插入操	作26	5
	2.4.1.	控制器后台的操作支持26	5
	2.4.2.	界面层的代码27	7
2.5.	删除操	作	L
	2.5.1.	控制器后台的操作支持31	L
	2.5.2.	界面层的代码34	1
2.6.	获取对	象	7
	2.6.1.	控制器后台的操作支持37	7
	2.6.2.	界面层的代码	3
2.7.	更新操	作40)

	2.7.1.	控制器后台的操作支持	41
	2.7.2.	界面层的代码	42
2.8.	字典数	据绑定	43
	2.8.1.	Select2 控件介绍	44
	2.8.2.	Select2 控件的实际使用代码分析	47
	2.8.3.	Select2 控件的赋值处理	50
	2.8.4.	控制器后台的操作支持	51
	2.8.5.	界面层的代码	52
	2.8.6.	下拉树形列表	53
2.9.	树信息	绑定	55
	2.9.1.	插件 JSTree	56
	2.9.2.	JSTree 插件的复选框处理	59
	2.9.3.	控制器后台的操作支持	63
	2.9.4.	界面层的代码	64
2.10.	文件上	传插件 FILE INPUT 的使用	66
	2.10.1.	文件上传插件 File Input 介绍	66
	2.10.2.	文件上传插件 File Input 的使用	68
	2.10.3.	bootstrap-fileinpu 实现数据导入处理	71
2.11.	文件上	传 UPLOADIFY 的使用(停用)	81
	2.11.1.	控制器后台的操作支持	82
	2.11.2.	界面层的代码	83
2.12.	附件管	理和图片预览、OFFICE 文档预览	85
	2.12.1.	控制器后台的操作支持	87
	2.12.2.	界面层的代码	89
	2.12.3.	利用微软的平台进行 Office 文档的在线查看	91
	2.12.4.	把 Office 文档生成 HTML 文件后进行查看	95
2.13.	EXCEL	数据导入导出	97

第 2页 共 149页

	0 10 1	按周期亡人的提供于社	100
	2.13.1.	控制器后台的 架 作文持	100
	2.13.2.	界面层的代码	103
2.14.	图标样	式生成和选择	106
	2.14.1.	菜单的显示及各种 Bootstrap 图标	107
	2.14.2.	各种 Bootstrap 的图标的提取	110
	2.14.3.	Bootstrap 的图标显示和选择	114
	2.14.4.	控制器后台的操作支持	116
2.15.	统计图	表 HIGHCHARTS 使用	118
	2.15.1.	Highcharts 基础介绍	119
	2.15.2.	控制器后台的操作支持	121
	2.15.3.	界面层的代码	122
2.16.	标签控	件 TAGS-INPUT	124
	2.16.1.	Tags Input 插件的使用	125
	2.16.2.	在项目中使用 Tags Input 插件	127
2.17.	RDLC	报表生成和预览	129
	2.17.1.	控制器后台的操作支持	129
	2.17.2.	界面层的代码	131
2.18.	LODO	?打印模块的使用	132
	2.18.1.	LODOP 打印控件介绍	132
	2.18.2.	使用 LODOP 打印控件	133
2.19.	富文本	控件 CKEDITOR 和 CKFINDER 控件	136
	2.19.1.	CKEditor 的使用	136
	2.19.2.	CKFinder 的集成使用	138
	2.19.3.	集成效果展示	139
	2.19.4.	MVC 的处理	141
2.20.	SUMMEI	RNOTE 插件实现 HTML 文档的编辑和图片插入	142
	2.20.1.	Summernote 的简单使用	142

2.20.2. summernote 插件在实际项目中的使用144

1. 引言

1.1.背景

这几年,我一直专注于开发框架的研究和应用,开发出来的产品,先后有《Winform开 发框架》、《WCF开发框架》、《混合式开发框架》、《基于 EasyUI 的 Web 开发框架》、《微信开 发框架》等框架以及一些项目的开发工作,这些框架技术在各自的领域发挥了极大的作用, 特别是框架的案例应用结合代码生成工具的快速开发,给客户提供一个统一、可重用、快速 高效的标准开发模式,为用户开发各种应用提供了快捷、稳定的平台和解决思路,同时也为 各个企业培养了很多技术人员,帮助企业积累了很多基础应用开发的知识和经验。

这几年,随着 JQuery 技术的兴起,很多技术应用如雨后春笋般冒出来,因此我的《基于 EasyUI 的 Web 框架》也随之产生,这个框架很好利用了 Ajax 和 Jquery 及相关控件等技术,结合微软的 MVC 后台处理技术,在开发管理后台界面实现了非常不错的展示效果。而通过和代码生成工具 Database2Sharp 的整合,可以快速开发后台代码以及基于 MVC 的控制器和前端视图代码,因此可以极大提高开发效率,以及统一系统模块的界面样式。

随着 Web 前端技术的发展,Bootstrap 响应式技术框架也随之出现,这种响应式的技术 框架,除了能够很好适应各种设备展示外,还提供了很多 CSS 的酷炫样式,在很多大企业以 及前端应用里面,采用 Bootstrap 技术的网站和平台也越来越多,基于 Bootstrap 技术开发 出来的插件也琳琅满目,因此可以结合开发出功能很强大的系统应用。Bootstrap 响应式技 术框架除了可以制作界面美观、功能强大的后台管理界面外,还可以做适应在小设备上的微 网站、购物商城等热门的 Web 应用。

基于技术研究和革新的目的,我们引入更新、更强大的 Bootstrap 和 Metronic 框架技术,整合之前的《基于 EasyUI 的 Web 开发框架》,继续使用成熟的 MVC + Enterprise Library 等相关的技术对原有的 Web 开发框架进行了升级。该《Web 开发框架》是我们经过多年的项目积累,吸收众多框架产品的前沿思路,以及客户的宝贵意见,反复提炼优化而成的。

1.2. 编写目的

本文档主要介绍《基于 Metronic 的 Bootstrap 开发框架》的基础上,通过分析各个模 块相关的代码、各种常见组件的使用、特殊应用场景的处理,熟悉掌握框架里面相关代码和 组件的使用,以便能够更全面的、更真实了解《Bootstrap 开发框架》(以下称《Bootstrap 开发框架》)的相关特点和如何使用框架的相关模块代码。

1.3. 参考资料

序号	名称	版本/日期	来源
1	《Web 开发框架-架构设计说明书.doc》		内部
2	《Winform 开发框架-架构设计说明书.doc》		内部
3	《循序渐进开发 Web 项目操作说明书.doc》		内部
4	《Bootstrap 开发框架-系统功能介绍白皮书》		内部

1.4. 术语缩写及约定

在本文件中出现的 "Bootstrap开发框架"一词,除非特别说明,均指基于MVC +
 Bootstrap + Enterprise Library等相关的技术研制的 Web开发框架。

2 在本文安装.NET框架中,除非特别说明,均指 .NET 4.5 框架。

3 本文的开发环境为Visual Studio 2013,基于 MVC5 的Web开发技术。视图HTML 代码编辑使用Sublime Text编辑工具。

2. 框架界面代码的分析

2.1. 视图界面代码介绍

在视图界面里面,各个模块都是以 DIV 的方式进行处理,增加、修改、查看、导入等 界面,都是有专有的 Div 层与之对应,这些部分再使用代码生成工具 Database2sharp 就可以 生成标准的界面了。

Database2Sharp[已)	主册]												- 0	
ኼ(S) 工具(T) 帮	助(H) 窗口(W)												
主工具栏视图 🦄 数	据库配置	alserver	r N	/ 🔔 重新加	载数据库 - 例 实	体类属性生成	È 🚺	Winform	。 界面代码 💋 Ea	svUI的Web界i	面代码 🐻 Boo	ostrap的Web界i	面代码 │ 🛛 退出	~
库信息	ΦX		主其大/合6		,		-							-
weixinApp	^	- 売空印	及至4165 日本信息	2										-
🖗 WangGou			A. 282-4-1147-026									易	噺 设置字則	段别
MemberMis		_	100	- × -	and the state and	Table Art rate and		La atr	1.66	100		RID L AND		_
ForumMis			编号	名称	字段类型	控件类型	_	长度	王確	目増	可空	默认值	别名	3
» CRM-HIS			0 1	.D	Int	数值类型	•	4	True	True	False		ID	4.
> EDNMS3			1 F	ID	Int	数值类型	•	4	False	False	True	-1	PID	_3
» EDNMS_Database			2 }	landNo	WarChar	单行文本	•	50	False	False	True		HandNo	Я
» HeartPatientMis			3 1	lame	WarChar	单行文本	•	50	False	False	False		Name	Я
• MVCWebMis			4 F	assword	NVarChar	単行又本	•	100	False	False	False		Password	H
MVCWebMisTest	(1) 77 (1 - 1)		5 F	ullName	WarChar	单行文本	•	100	False	False	True		FullName	ţ
Tables	代码生成		•	Ente	erpriseLibraryf	℃码生成(E)			False	False	True		Nickname	月
T_ACI	实体类生成快速	速入口	•	▶ Web界面代码生成(1)					False	False	True	0	IsExpire	
T_ACI	实体类属性生	成(P)		Bootstrap的Web界面代码生成(B)					False	False	True		Title	E
T_ACI	Winform界面	代码生质	龙(W)	Enti	tvFramework@	体框架代码件	= ह रि/(F		False	False	True		IdentityCard	1 3
T_ACI	数据库文档生	成(D)			,			<u> </u>	False	False	True		MobilePhone	1
T_ACI	SOL 查询分析	88 (Δ)		We	b API控制器代码	4生成(W)		_						
T_ACI		(MA)		ABLE [d	bo].[T ACL	User] (
T_ACI		(191)	, , , , , , , , , , , , , , , , , , ,											
I_ACI	拷贝列农内容	(C)		Int N	OT NULL IDE	NTITY (1,	1)	,						
T_ACI	表别名修改			Int N	ULL DEFAU	LT (-1) ,								
T_ACI	重新加载数据	库(R)			Char(50)	NULL ,								
T_ACI	剧 新粉 捍 左 利:	#		sword]	NVarChar(10	0) NOT N	ULI							
T ACL USer	////www.goug/=//	19	[20	IName]	NVarChar(10	0) NULL	,							
T_ACL_User	_Role	10	[Ni	.ckname]	NVarChar(10	0) NULL	,							
T_CRM_Comp	etitor	11	[Is	Expire]	Bit NULL	DEFAULT (C))	·						
T_CRM_Cont	act	12	[Ti	tle] NVa	rChar(100)	NULL ,	-							
T CRM Cont	actGroup	13	[10	bilePhon	el NVarChar	(100) NUT	ulu T.	'						
T_CRM_Cust	omer	15	[Of	ficePhon	el NVarChar	(100) NUI	L	<i>.</i>						
T_CRM_Cust	omerGroup	16	[Ho	mePhone]	NVarChar(1	.00) NULL		,						
T_CRM_Cust	omerGroup_	17	[En	ail] NVa	rChar(100)	NULL ,								
⊡ I TB Address	> `	18	[Ad	ldress] N	VarChar(255) NULL	,							
	-	10	[We	rkAddrl	NVarChar(25	5) NULLT.								

一般来说,对整个业务操作的界面,我们都放在 index.cshtml 的视图文件里面,这个文件分为了各个部分,包含视图定义、引入 JS、查询层、列表展示层、添加修改层、查看信息层、导入数据层等等。



其中,整个视图是基于布局主界面 "~/Views/Shared/_Layout.cshtml",在_Layout.cshtml

页面里面,通过@ViewBag.Title 来呈现页面标题,通过 Render 来展现更多子页面的内容,

如下所示。



2.2. 框架布局设置

在常规的后台管理系统或者前端界面中,一般都有一个导航菜单提供给用户,方便选择 所需的内容。本框架提供三种方式的菜单布局方便选择,两种左侧竖放的菜单,一种是水平 横放的菜单。

样式 1:



菜单收缩后界面如下所示。





功能菜单							
命 首页	~	系统田立信息					
● 行业动态	<	示现用广告志					
■ 会员管理		▼ 用户分类	▼杳	询内容			
■ 客户管理	<	按组织和构态差 按角色态差					
合 权限管理	< <			用户编	码		
▲ 权限管理	<			移动电]话		ŧ
& 用户管理		→ 響 总经办					
🔲 组织机构管理		- 警 工程部					
A. 角色管理		□- 警 产品研发部	08	据列表			
🙈 系统类型定义		- 开发一组					
읍 功能管理			每页	显示 10	▼ 条记录	共有记录: <mark>461</mark> 条,)	总页数: <mark>47</mark> 页。
⊨ 菜单管理		□ 響 市场部					
■ 登陆日志				ID	用户编码	用户名/登录名	真实姓名
	<			462	shadmin	shadmin	上海管理员
		- 響 生产部		463	bjadmin	bjadmin	北京管理员
				464	gzadmin	gzadmin	广州管理员
		│		18	100001	100001	陈莉
				93	100004	100004	邓卉
		─ 雪 炒分節		145	100006	100006	黄晓雪

样式 3:

水平样式的菜单如下所示:

METRONIC				?) 4	🐏 🧖 管理员 🗸
● 行业动态 🛛 🔮 会员管理 📲	🖻 客户管理 🛛 🔓 权限管理	✿ 系统管理			
字典信息		 \$\$\$ \$\$\$ \$\$\$\$ \$\$\$ \$\$ <li< th=""><th>> ></th><th>■ 字典管理■ 全国省份</th><th></th></li<>	> >	■ 字典管理■ 全国省份	
▼ 字典类别 +新2 2 680 - 删除	▼ 查询内容			📮 全国城市 🗅 全国行政区	國學出 🗡
□- · · · · · · · · · · · · · · · · · · ·	字典名称		备注	▲ 图标管理	
- · · · · · · · · · · · · · · · · · · ·	c\$ 数据列表 每页显示 10 ▼ 条记录	+批里添加 t 共有记录: 458 条	+ 新增 ₹,总页数:	✔修改 冊查香 一 册 46 页。	略注 ご 刷新 こく
	 字典名称 使用位置 	字典值	备注 排序 0	编辑时间 2011-02-27 14:11:31	操作 Q 🖊 🗙
── ➡ 负责需求	□ 标准件(国产)	标准件(国产)	0	2013-09-22 17:00:45	

由于这几种布局方式,它们的框架布局页面(也就是 Layout 页面)内容有所不同,一般情况下我们可以把它分别定义为一种 Layout 页面,这样在实际使用的时候包含它即可。

我根据上面几种样式,分别定义了几个不同的 Layout 页面,如下所示



一般情况下,我们生成一个视图页面,会提示我们选择不同的视图布局页面的,确认之后会 在页面顶部应用对应的 Layout 页面。

index.cshtml 😕 🗙
@{
Layout = "~/Views/Shared/_Layout.cshtml";
ViewBag.Title = "角色信息";
3
@*脚本引用放在此处可以实现自定义函数自动提示*@
<script src="~/Scripts/CommonUtil.js"></script>
□ <div class="portlet box green-meadow col-md-3"></div>
<pre>div class="portlet-title"></pre>
div class="caption">
<i class="fa fa-filter"></i> 角色列表
div class="tools">
<pre>cbutton type="button" class="btn btn-circle btn-sm green-meadow" onclick="ShowImport()"></pre>
<i class="fa fa-file-excel-o"></i>
し
// button>
<pre>display="button" class="btn btn-circle btn-sm green-meadow" onclick="ShowExport()"></pre>
<i class="fa fa-file-excel-o"></i>
ゆうしょう ゆうしょう ゆうしょう しょうしょう しょう
// Alton>
<pre>display="button" onclick="Refresh()" class="btn btn-circle btn-sm green-meadow" data-toggle="modal">btn-circle btn-sm green-meadow" data-toggle="modal">btn-circle btn-sm green-meadow" data-toggle="modal">btn-circle btn-sm green-meadow</pre>
<i class="fa fa-refresh"></i>
刷新
/button>

如果我们需要动态指定页面的布局内容,可以在上面使用一个变量来处理,如下代码所示。

```
@{
   Layout = ConfigData.ViewLayoutFile;// "~/Views/Shared/_Layout.cshtml";
   ViewBag.Title = "角色信息";
}
```

实际上,我们一般的页面布局确定好后,就相对比较少变更的,所以为了不打破这个生成页面的布局内容,我们可以把前面处理好的_Layout1/2/3等这些页面内容复制到 _Layout.cshtml页面里面进行处理即可。



2.3. 分页查询

在很多场合,分页是必须的,一个是可以节省响应时间,二是提高数据检索效率,使得 我们能够给系统界面提供更好的用户体验。

在 Winform 开发框架里面,我通用把分页部分独立做为一个分页控件,那样给用户提供更好的一致的界面,而且也能封装一些常规的菜单处理。不过在 MVC + Bootstrap 界面处理方面,这个有所不同,为了很好支持分页处理,我们使用相应的分页插件进行业务数据的分页处理。

我们引入了 Bootstrap 的插件 bootstrap-paginator 插件和 Bootstrap-Table 插件进行分页 处理,通过在视图 HTML 界面的 JS 代码里面进行一定的配置,另外 Web 框架的控制器基 类也进行了相应的封装,这样就可以完美实现 Bootstrap 框架中的条件分页处理了。

1) 常规 bootstrap-paginator 分页方式

最开始的 Web 界面列表分页,使用较为常规 Bootstrap Paginator 分页模式,内容 生成以 HTML 组合方式,先设置表头,然后获取分页列表数据,遍历生成相关的 HTML 数 据,增加到页面上,这种方式也是比较高效的处理方式。下面是框架里面分页部分的效果图。

查ì	旬内容		条件查i	询区域 📜			٩	查询 🔄 导入	、 🕑 🖶	<u>ж)</u> (】 RDLC报	表
	用户编	码		я	沪名			真实姓名				
	移动电	话		邮件	地址			QQ号码				
数	据列表	• 5 113			页码明细)	+ 新増	✔ 修改 田 董	iā — H	刊II余	€ 刷新	
1.416	小 <u>10</u>	• <u></u>	田白々 成果々	直安县々	县不过期	星本加級	数据显示	示区域	하는 우리		18.45	
	462	shadmin	shadmin	上海管理员	正常	正常	134)14	叫什小巴加	11177月			e 1
	463	bjadmin	bjadmin	北京管理员	正常	正常			男		Q /	•
	464	gzadmin	gzadnin	广州管理员	正常	正常			男		Q /	e :
	18	100001	100001	陈莉	正常	正常	13810107123		男		۹,	e :
	93	100004	100004	邓卉	正常	正常	13917028898		女		۹.	e :
	145	100006	100006	黄晓雪	正常	正常	18674042588		女		۹.	0
		100000	100009	王景景	正常	正常	13761601760		男		Q,	e :
	350	100009										
	350 53	100003	100012	朱丽华	正常	正常	13661246623		男		۹.	•
	350 53 438	100012	100012	朱丽华王伟	正常	正常	13661246623		男男	•	a .	e :

这种方式可控性非常好,而且可以对 HTML 代码进行完全的控制,非常适合在自定义 界面中展示一些数据,如我之前介绍过的图标分页展示界面一样,完全是自定义的内容展 示,图标界面如下所示。

▼ 图标分类	▼图标选择	示例									
○班新生成即标【删除后生成】 ☐ ☐ FontAwesome ☐ ☐ Glyphicons ☐ SimpleLine	Web图标 Icon-screen-tablet 选择图标 Default Primary Success Info Warning Danger 显示名称 显示名称 重查询										
	▼ 图标信息 毎页显示 50	▼ 条记录	共有记录:()条,总页数;	: <mark>0</mark> 页。						
	÷	Ð	\triangleright		Ø	Q	Ů	Ĥ	$\langle $	占	<u>م</u>
	Ø	ŵ	<u>D</u> ,	$\langle \rangle$		A	X	\bowtie	P	2	
	Ø		C	\mathcal{Q}	Ŷ	\bigtriangledown	X	\odot	Ø	٩٩	Ŷ
	Ē	\sim	1	+- x=	j	Q		000	ର୍ବ	\sim	÷
		ŝ	O [™]		₿	Ś					
	1 2 3	3 4 >	>>								

2) Bootstrap-table 分页方式

随着开发项目的需求变化,对数据记录分页展示、排序等功能都是常态的要求,因此在

第 13页 共 149页

代码生成工具中调整了主列表界面的列表展示插件为 Bootstrap-table 插件。

使用常规的分页方式界面可控性非常方便,不过随着不同项目的一些特殊要求,对表头 排序的需求也是非常强烈的,上面的分页处理方式无法实现表头的排序功能,因此引入了使 用非常广泛的 Bootstrap-Table 插件,该插件应用很广、功能非常强大,可以通过属性配 置实现很细致的功能控制。Bootstrap-table 插件提供了非常丰富的属性设置,可以实现查 询、分页、排序、复选框、设置显示列、Card View 视图、主从表显示、合并列、国际化 处理等处理功能,而且该插件同时也提供了一些不错的扩展功能,如移动行、移动列位置等 一些特殊的功能。这个插件界面展示也是非常美观的。

▼查	间内容						Q查询	國导入 國导出 🗸
客户约	编号	客户名称		客户简称			公司地址	
《数	据列表				+	・新増 🥜 修改	田查看 -	一部除 2 刷新 王
								₽
	客户编号 🍦	客户名称	客户简称	客户类别	信用等级	公开与否 🍦	客户状态	操作
	KH1010	中国电子科技集团公司第三十四研究所	三十四研究所	医院	一般	不公开	跟进	a 🖉 🔀
	KH1009	宁波赛柯国际贸易有限公司	赛柯	流动商	一般	不公开		Q 🖍 🗙
	KH1008	深圳市鵬海运电子数据交换有限公司	鵬海运	流动商	良好	不公开	合作	Q 🖉 🗙
	KH1002	内蒙古医科大学第二附属医院	内蒙古	正式客户	优秀	不公开	潜在	Q 🗷 🗙
	KH1007	浙江迦南科技股份有限公司	迦南	潜在客户	一般	不公开	跟进	Q 🖉 🗙
	KH1005	北京嘉信仕通电子科技有限公司	嘉信仕通	正式客户	良好	不公开	合作	Q 🖉 🗙
	KH1006	南京军区南京总医院	南总	潜在客户	一般	不公开	跟进	Q 🖉 🗙
	KH1004	银川嘉泰神机工程软件技术有限公司	嘉泰神机	正式客户	一般	不公开	潜在	Q 🗷 関
	KH1003	南方医科大学南方医院	南方医院	合作伙伴	一般	不公开	合作	Q 🖉 🔀
	KH1001	广州爱奇迪软件科技有限公司	爱奇迪	流动商	优秀	不公开	合作	Q 🖉 🗙
显示	第 1 到第 10 条记	渌,总共 10 条记录					广州爱者 http://ww	所迪 伍华聪 ∕w.iqidi.com—

这个插件最显著的特点就是完美支持客户端或者服务器的数据列排序处理,单击表头就 可以实现排序操作。

上面的图示里面,整个界面,包含了查询区域以及数据列表两部分,其中数据列表里面, 包含了页面明细内容、数据列表显示以及页面区域部分,从而构成了一个标准的分页界面显 示,下面我们来详细分析下这个界面的实现思路和内在逻辑的处理。

2.3.1. 控制器后台的分页支持

Bootstrap 开发框架沿用了我的《<u>Winform 开发框架</u>》和《<u>基于 EasyUI 的 Web 框</u>架》</u>的很多架构设计思路和特点,对 Controller 进行了封装。使得控制器能够获得很好的继承关系,并能以更少的代码,更高效的开发效率,实现 Web 项目的开发工作,整个控制器的设计思路如下所示。



在 MVC 的控制器基类里面,BusinessController<B,T>类封装了增删改查等大多数通用的处理函数,里面就包括了分页的控件处理,下面是这个类的总体介绍。



下面是控制器分页的主要逻辑代码



上面的代码里面,控制器根据前端分页插件传回来的参数,构造好分页条件和分页信息,

代码如下所示。



而其中获取分页参数(也就是当前是几页,每页显示几行)。



2.3.2. 界面的条件内容

框架的列表界面里面,都有一个查询的条件区域,就是允许用户输入一定的条件进行数据的过滤查询,界面如下所示。

▼ 查询内容		Q 查 询	■ 号 入	3 号 出	武 RDLC报表	~
用户编码	用户名		真实姓名			
移动电话	邮件地址		QQ号码			

其中查询页面的 HTML 代码是通过代码自动生成的,我们开发的时候,可以根据实际 需要进行一定的调整,如去除部分不必要的查询条件,尽可能保持简洁、必要的查询字段即 可,一般建议不超过两行。



上面代码里面,每个 Div 的 class 都使用 class="col-md-4 col-sm-4 col-xs-6",这种是 Bootstrap 的栅格系统布局的声明。lg 是 large 的缩写,md 是 mid 的缩写,sm 是 small 的缩 写,xs 是***的缩写。这样命名就体现了这几种 class 适应的屏幕宽度不同。

	超小屏幕设备 手机 (<768px)	小屏幕设备 平板 (≥768px)	中等屏幕设备 桌面 (≥992px)	大屏幕设备				
栅格系统行为	总是水平排列	开始是堆叠在一起的,超过这些阈值将变为水平排列						
最大 .container 宽度	None (自动)	750px	970px	1170рх				
class前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-				
列数	12							
最大列宽	自动	60px	78px	95px				
槽宽	30px (每列左右均有15px)							
可嵌套	Yes							
偏移 (Offsets)	Yes							
列排序	Yes							

列偏移:使用.col-md-offset-*可以将列偏移到右侧。这些 class 通过使用*选择器将所有列增加了列的左侧 margin。例如,.col-md-offset-4将.col-md-4向右移动了4个列的宽度。

嵌套列:为了使用内置的栅格将内容嵌套,通过添加一个新的.row 和一系列.col-md-*列到已 经存在的.col-md-*列内即可实现。嵌套 row 所包含的列加起来应该等于 12。

列排序:通过使用.col-md-push-* 和 .col-md-pull-*就可以很容易的改变列的顺序。

col-md-"用法										
.col-md-1 .col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8							.col-md-4			
col-sm-*用法										
.col-sm-8							.col-sm-4			
col-xs-"用法										
.col-xs-8							.col-xs-4			
列偏移: col-md-offset-"										
.col-md-4							.col-md-4 .col-md	-offset-4		
		.col-md-3 .col-md	I-offset-3					.col-md-3 .col-md	d-offset-3	
		.col-md-6 .col-md	I-offset-3							
嵌套列: 嵌套row所包含的列加起来应	该等于12									
Level 1: .col-md-9										
Level 2: .col-md-6			Level 2	2: .col-md-6						
列排序: .col-md-push-*和.col-md-pu	11-2									
.col-md-3 .col-md-pull-9		.col-md-9 .col-md	I-push-3							

2.3.3. 界面的分页代码

在上面的分页处理函数里面,FindWithPager函数提供了对条件的分页处理,这个函数通 过条件获取指定页码的数据,然后返回一个JSON格式的数据给视图界面处理,并通过脚本 进行绑定到相应的插件上即可。

文件名称: Web 框架界面代码分析



上面的函数是初始化分页控件,我们可以在页面完成加载后,进行分页数据的显示操作,

如下代码就是进行数据的显示。



第 20页 共 149页

数据的显示,是在页面准备完成后,就是通过 Search 脚本函数进行处理,处理的时候, 先序列化表单的条件和分页的条件信息。

var condition = \$("#ffSearch").serialize();

然后传入 MVC 控制器,获取对应的列表数据,在界面上进行动态绑定即可完成整个处理过程了。具体代码截图如下所示。

<pre>var \$table;</pre>									
<pre>var queryUrl;</pre>									
//初始化bootstrap-table的内容	//初始化bootstrap-table的内容								
function Search (page) {									
//记录页面bootstrap-table全局变量\$table,方便应用									
<pre>queryUrl = '/Customer/FindWithPager?rnd</pre>	=' + Math.random()								
<pre>\$table = \$('#grid').bootstrapTable({</pre>									
url: queryUrl,	//请求后台的URL(*)								
method: 'GET',	//请求方式(*)								
<pre>//toolbar: '#toolbar',</pre>	//工具按钮用哪个容器								
striped: true ,	//是否显示行间隔色								
cache: false ,									
//是否使用缓存,默认为true,所以一般	B情况下需要设置一下这个属性(*)								
pagination: true ,	//是否显示分页(*)								
sortable: true ,	//是否启用排序								
sortOrder: "asc",	//排序方式								
<pre>sidePagination: "server",</pre>	//分页方式: client客户端分页, server服务端分页(*)								
pageNumber: page,	//初始化加载第一页,默认第一页,并记录								
pageSize: rows,	//每页的记录行数(*)								
pageList: [10, 25, 50, 100],	//可供选择的每页的行数(*)								
search: false ,	//是否显示表格搜索								
strictSearch: true ,									
showColumns: true ,	//是否显示所有的列(选择显示的列)								
showRefresh: true ,	//是否显示刷新按钮								
minimumCountColumns: 2,	//最少允许的列数								
clickToSelect: true ,	//是否启用点击选中行								
//height: 500,									
//行高,如果没有设置height属性,表格	各自动根据记录条数觉得表格高度								
uniqueId: "ID",	//每一行的唯一标识,一般为主键列								
showToggle: true ,	//是否显示详细视图和列表视图的切换按钮								
cardView: false ,	//是否显示详细视图								
detailView: false ,	//是否显示父子表								
//得到查询的参数									
queryParams : function (params) {									

columns:	[{						
check	oox: true.						
visibl	le: true	//是否显示复选框					
} .							
{ title:	'客户编号',	field: 'HandNo' /*, width: 80, sortable: true */ },					
{ title:	'客户名称',	field: 'Name' /*, width: 80, sortable: true */ },					
{ title:	'客户简称',	<pre>field: 'SimpleName' /*, width: 80, sortable: true */ },</pre>					
{ title:	'所在省份',	<pre>field: 'Province' /*, width: 80, sortable: true */ },</pre>					
{ title:	'城市', fie	<pre>ld: 'City' /*, width: 80, sortable: true */ },</pre>					
{ title:	'所在行政区	', field: 'District' /*, width: 80, sortable: true */ },					
{ title:	'市场分区',	<pre>field: 'Area' /*, width: 80, sortable: true */ },</pre>					
{ title:	'公司地址',	<pre>field: 'Address' /*, width: 80, sortable: true */ },</pre>					
{ title:	'公司邮编',	<pre>field: 'ZipCode' /*, width: 80, sortable: true */ },</pre>					
{ title:	'办公电话',	<pre>field: 'Telephone' /*, width: 80, sortable: true */ },</pre>					
{ title:	'传真号码',	<pre>field: 'Fax' /*, width: 80, sortable: true */ },</pre>					
{ title:	'主联系人',	<pre>field: 'Contact' /*, width: 80, sortable: true */ },</pre>					
{ title:	'联系人电话	', field: 'ContactPhone' /*, width: 80, sortable: true */ },					
{ title:	'联系人手机	', field: 'ContactMobile' /*, width: 80, sortable: true */ },					
{ title:	'电子邮件',	field: 'Email' /*, width: 80, sortable: true */ },					
{ title:	'QQ号妈', f.	ield: 'QQ' /*, width: 80, sortable: true */ },					
{ title:	'答尸奕别',	field: 'CustomerType' /*, width: 80, sortable: true */ },					
title:	'各尸级别',	field: 'Grade' /*, width: 80, sortable: true */ },					
{ title:	'信用寺级',	field: 'CreditStatus' /*, width: 80, sortable: true */ },					
title:	'里安级别',	<pre>field: 'importance' /*, width: 80, sortable: true */ },</pre>					
{ title:	"公开'习'行",	field: 'Ispublic' /^, Width: 80, sortable: true // },					
(title:	"各厂则权",	field: Stage /^, Wight: 80, Softable: true // },					
title:	一谷广 小心 ,	Left Status /*, with 00, softable the */ },					
{ title.	各广州周八	A, Heid, Dolated /*, with 00, sortable, true */ },					
title.	·撮作! fie	, Held. Deleted /~, Width. 60, Soltable. the ~/],					
1 01010.	1741 P , 110.	a. ib, witch. 120, align. center, valign. mittie, formatter. attomformatter,					
onLoadSucc	ess: functi						
currer	ntPage = nag	2.//左佬当前页码					
}.	iorago - pag						
onLoadErro	or: function	0.1					
showTi	ips("数据加载	这失败! "):					
} .							
onDblClic	onDblClickRow: function (row, Selement) (
var ic	d = row.ID;						
EditVi	iewById(id,	'view');					
EditVi	iewById(id,	'view');					

而其中的代码: { title: '操作', field: 'ID', width: 120, align: 'center',

valign: 'middle', formatter: actionFormatter },

则是通过脚本生成一些操作按钮,

```
//提作E的格式化
function actionformatter(value, row, index) {
var ids value;
var ids var ids var ids var ids var ids var ids v
```

生成界面如下所示。

操作
Q 🖉 関
Q 🖉 関
Q 🖉 関
Q 🛛 関

2.3.4. 在代码生成工具中集成分页支持

我们的代码生成工具 Database2Sharp 是为了框架开发服务的,不管是 Winform 还是 Web

第 22页 共 149页

开发,都可以基于数据库的基础上进行框架代码的快速生成,以及界面的代码生成,本次调整的代码生成工具功能,在列表界面代码中增加了对 Bootstrap-table 插件分页的支持,使得我们开发 Bootstrap 框架的界面代码更加丰富、快捷。

在代码生成工具 Database2Sharp 上,我们先使用 Enterprise Library 代码增量生成主体框架的框架代码。

🎒 Database2Sharp[Ē	3注册]									
ぞ体(の) 丁目(の)	±setrun as⊡ru	n								
系统(5) 工具(1)	帮助(H) 图口(V	v)						_		_
: 🎯 主工具栏视图 🤻	数据库配置 sq	server	`	_ 😂	重新加载数	据库 分实体	美属性生成 🥻	7 W	/inform界面代	跨 👩 Ea
数据库信息	Ψ×	/ 💷 🕯	表基本信	息						
🗄 🌓 CarDispatchMis	^	表字段	基本信息	1						
🗄 📑 ClothingMis										
HealthWis			伯口	わび	2	今の米田	1金/生业4月1		と 庇	
🗄 🕞 HeartPatientMis	5		細ち	石利	1	子校尖型	控件类型	_	大反	土珽
🕀 🛑 Hotel			0	ID		NVarChar	单行文本	•	50	True
🗄 🌗 EmailPorter			1	Hand	No	NVarChar	单行文本	•	50	False
⊞			2	Name		tWarChar	单行文本	•	255	False
E Tables	代码生成		•		Enterpris	。 seLibrary代码生	_成(E)			False
	实体类生成快速	λΠ	•		Web界面	代码生成(1)		-	_	False
T_AC	空休券届性生成((P)			Pootstra		교수 라(P)			False
T_AC		с) Пан-там	`		BOOISITA	ризмеранци			_	False
	Wintorm赤山(V	19±0%(VV)		EntityFra	mework实体框	架代码生成(F)			False
	数据库文档生成((D)				按制器代码生成				False
T_AC	SQL 查询分析器	(A)		атре	oue	This at cust	(***) 千11 × 4		90	False
 T_A(SQL语句生成(M)	Þ	Tele	phone	WarChar	单行文本	•	50	False
	拷贝列表内容(C)			44.00	Timel Da	toTime NUL	T			
T AC				ele	tedl Int	NULL	L ,			
T_AC	农别名1000			ept	ID] NVa	rChar(50)	NULL ,			
T_AC	重新加载数据库((R)		omp	any ID]	NVarChar(5	0) NULL	,		
T_AL	刷新数据库列表			NST	RAINT [F	K_T_CRM_Cu	stomer] P	RIN	ARY KEY CI	USTERE
	or	163 7								
T_ACL_Us	er_Role	64								ner et ver
T_CRM_Co	mpetitor	65	exec s	p_a	adextend	leaproperty	N'MS_Des	cri	ption', N'	· 杏厂 编 · 安 白 夕
T CRM Co	ntact	66 6	exec s	p_a	adextend	eaproperty	N.W2_Des	cri	peron, N.	27.4

然后在使用 Bootstrap 的 Web 界面代码生成功能,如下可以在工具栏界面中选择。

5(S) 工具(T) 帮助(H) 窗口(V	V) Iserve	r v 😒	重新加载数据库 必定(1.送属性生成「	Ninform馬	面代码 🜈 Easy	UI的Web界面代		p的Web界面	武代码 🛛 退出 🔿	െ
车信息 平 ×		- 表基本信白									۰.
CarDispatchMis 🔨	表字	2段基本信息									
> ClothingNis										刷新 设置;	字段另
> EDNMS_Database											-
> RealthMis		编号名科	『 字段类型	控件类型	长度	主键	日増	可空	默认住	ē 别名	1
Hotal	Þ	0 ID		单行文本	▼ 50		False				
EnailPorter		1 Hand	No NVarChar	单行文本	▼ 50	Fal	False	True		HandNo	
MVCWebMiz		2 Name	Wavehav	前行立本	• 255	and an	Falsa	Terro		Name	2
⇒ MVCWebMisTest		2 Indie						- 0	×	ci-l.x.	7
- Tables	- 11	5 51mj	1 法法权 四丁 根 約 約 由 5	e ha logi					~	Simpremane	12
T_ACL_BlackIP		4 Prov	11 頃応挥用丁傑[日]次5	5490BC						Province	B
		5 City					v a .			City	ţe
- I_ACL_Function		6 Dist	ri TALL_LoginLog			^ I_UK	M_Customer		^	District	戶
T_ACL_LoginLog		7 Area	T_ACL_Operation	og						Area	ĥ
T_ACL_Menu		8 444	T_ACL_Operation	logSetting		>>				Address	1
T_ACL_OperationLog		0 7/-0	T_ACL_OU_Role							71-0-1-	
T_AUL_UperationLogSet	- 11	5 L1pt	T_ACL_OV_User							Lipcode	2
		10 Tele	pF I_ACL_Nole T_ACL_Role Func:	ion						Telephone	戎
T ACL OU Man		(The second	T_ACL_RoleData								
T ACL DO USER	58	(Earc	I	lor1ze							
T ACL Bole Function	59	[Dere	T_ACL_User	•							
T ACL BoleData	60	[Dept	T_ACL_User_Role								
T ACL Systemäuthorize	61	[Comp	T CEM Contact								
T ACL SystemType	62	CONSI	T_CEM_ContactGr	up							
T_ACL_User	63)	T_CEM_ContactGr	oup_Contact							
T_ACL_User_Role	64		T_CRM_CustomerG	oup_Customer							
T_CEM_Competitor	65	exec sp_a	DC TB_Address							N'I_CRM_Custome	r,
	66	exec sp_a	TB AddressGroup	Address						N'T_CRM_Custome	er ; ,
T_CHM_ContactGroup	67	exec sp_a	TB_AppChannel	-		~			~	N'I_CRM_Custome	-r /
T_CEM_ContactGroup_Co	68	exec sp_a	CIC TR Rootetronian		>	<			>	N'T_CRM_Custome	er',
T_CEM_Customer	69	exec sp_a								CRM_Customer',	N'C
T_CBM_CustomerGroup	70	exec sp_a	0.0							, N'T_CRM_Custo	omei
T_CHM_CustomerGroup_C	71	exec sp_a	dd		_					N'T_CRM_Custome	er ,
TB_Address	72	exec sp_a	ac			取消 [.	上一步 ['	下一步	完成	N'I_CRM_Custome	er ,
TB_AddressGroup	73	exec sp_a	dd							N'T_CRM_Custome	er',
IB_AddressGroup_Addre	74	exec sp_a	ddextendedpropert	y N'MS_Des	scription'	, N'办公电话	', N'user',	N'dbo', N'	'table',	N'T_CRM_Custome	er',
>	75	exec sp_a	ddextendedpropert	y N'MS_Des	cription'	,N'传具专供	', N'user',	N'dbo', N'	'table',	N'T_CRM_Custome	er',

选择数据库和表后,可以进行界面代码(包括控制器代码、视图界面代码)两部分,其中视 图分为两种模式,一种是利用 Bootstrap-table 插件的分页及排序(index.cshtml),一种是常 规的 Bootstrap Paginator 分页处理(index2.cshtml)。

뮾 (C:)	> output > WHC >	TestProject > MVCWe	ebUI > Views > 0	Customer	
^	, 名称	` I	修改日期	类型	大小
	lindex.cshtml lindex2.cshtml	两种分页模	2017-09-24 11:52 2017-09-24 11:52 式下的视图	ASP.NET Web P ASP.NET Web P	
	(

老客户可以继续使用 index2.cshtml 的样式,也可以使用最新的 Bootstrap-table 插件的分 页及排序方式(index.cshtml)。

生成的界面分为 HTML 部分和 JS 部分,都是比较紧密联系的两部分,我们进行一定的 调整即可实现丰富的界面排版。

1	0(
2	Layout = "~/Views/Shared/ Layout.cshtml";
3	ViewBag.Title = "客户基本资料";
4	}
5	
6	U*脚本匀用放住瓜处可以头现目走入图数目动提示*U
0	<script src="~/scripts/commonutil.js"></script>
8	al protection 粉板本海 、
9	<pre>cite Bbgin Write 0->></pre>
10	HCAIV Class-"portiet by green-meadow">
±0 10	(DWD 次434 年 10]>
50	BEGIN 表格粉据
51	Rediv class="portlet box green-meadow">
91	
92	
93	E <div <="" aria-labelledby="myModalLabel" class="modal fade bs-modal-lg" id="add" role="dialog" tabindex="-1" th=""></div>
	aria-hidden="true">
77	
78	
79	E <div <="" aria-labelledby="myModalLabel" class="modal fade bs-modal-lg" id="view" role="dialog" tabindex="-1" th=""></div>
	aria-hidden="true">
11	
12	
13	E <div <="" aria-labelledby="myModalLabel" class="modal fade bs-modal-lg" id="import" role="dialog" tabindex="-1" th=""></div>
	aria-hidden=" true ">
46	

部分的 JS 代码(展示分页部分处理)如下所示。



列表数据的显示列,默认是以数据库的字段进行生成,我们在生成后可以进行一定的调整,可以合理显示我们关注的数据。

	.,	
columns:	1	
checki	oox: true,	八月不月二何选佐
VISIO	le: crue	//定百亚小夏远阳
<i>I</i> , +i+lo:	1 安白编具1	field, WandWe! (* width, 20 contable, true */)
t title.	百万 ¹ 一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	field. Name' /*, width: 80 sortable. true */ },
I title.	·安白简称!	field: 'SimpleName' /* width 80 sortable true */ }
{ title:	'所在省份'	field. 'Province' /* width: 80, sortable: true */ }
{ title:	'城市' fiel	d. [City] /*, width, 80, sortable, true */ }
{ title:	'所在行政区'	field 'District' /*. width 80. sortable true */ }
{ title:	'市场分区'.	field: 'Area' /*. width: 80. sortable: true */ }.
{ title:	'公司地址'.	field: 'Address' /*, width: 80, sortable: true */ }.
{ title:	'公司邮编'.	field: 'ZipCode' /*, width: 80, sortable: true */ }.
{ title:	'办公电话',	field: 'Telephone' /*, width: 80, sortable: true */ },
{ title:	'传真号码',	field: 'Fax' /*, width: 80, sortable: true */ },
{ title:	'主联系人',	<pre>field: 'Contact' /*, width: 80, sortable: true */ },</pre>
{ title:	'联系人电话'	, field: 'ContactPhone' /*, width: 80, sortable: true */ },
{ title:	'联系人手机'	, field: 'ContactMobile' /*, width: 80, sortable: true */ },
{ title:	'电子邮件',	<pre>field: 'Email' /*, width: 80, sortable: true */ },</pre>
{ title:	'QQ号码', fi	eld: 'QQ' /*, width: 80, sortable: true */ },
{ title:	'客户类别',	<pre>field: 'CustomerType' /*, width: 80, sortable: true */ },</pre>
{ title:	'客户级别',	<pre>field: 'Grade' /*, width: 80, sortable: true */ },</pre>
{ title:	'信用等级',	<pre>field: 'CreditStatus' /*, width: 80, sortable: true */ },</pre>
{ title:	'重要级别',	<pre>field: 'Importance' /*, width: 80, sortable: true */ },</pre>
{ title:	'公开与否',	<pre>field: 'IsPublic' /*, width: 80, sortable: true */ },</pre>
{ title:	'客户阶段',	field: 'Stage' /*, width: 80, sortable: true */ },
{ title:	'客户状态',	field: 'Status' /*, width: 80, sortable: true */ },
{ title:	'各尸所属人身	<pre>[', field: 'OwnerUser' /*, width: 80, sortable: true */ },</pre>
{ title:	'定台巳删陈'	, field: 'Deleted' /*, Width: 80, sortable: true */ },
{ title:	'∰%îF', Ilel	d:'ID', Width: 120, Aligh: 'Center', Valigh: 'Middle', formatter: actionFormatter},
],		
onLoadSucc	tDago - page	
l Currer	itrage = page	
J,	r: function	
showTi	ns("数据加裁	
1	Pol skuluitad	
onDblClick	Row: functio	n (row, Selement) (
var ic	d = row.ID:	
EditVi	lewById(id, '	view');
var 10 EditVi	d = row.ID; lewById(id, '	view');

当然生成的界面代码还有很多其他的 JS 代码,如编辑、查看的代码和控件对应,导入、 导出等代码的处理,都是一并生成的,我们根据需要进行一定的裁剪调整即可完成整个界面 的处理了,极大的提高开发效率。

2.4. 插入操作

在一个业务系统里面,除了分页,增删改查是必须要有的常规操作,这些操作对各个业 务对象来说,基本上逻辑都一样的,只是它们传递的数据不太一样而已,因此它们往往也是 放在基类进行定义,子类根据不同的需要重写一下函数就可以了。

在 Web 框架里面,我们不仅在业务层基类对这些常规的操作进行了封装,而且在控制器基类的层面,也对这些常规的操作进行了统一封装处理,基本上我们在控制器子类不需要额外的代码就能处理这些增删改查的操作了,实现一个业务的增删改查是非常方便快捷的。

2.4.1. 控制器后台的操作支持

在控制器后台,对插入操作进行了统一的封装,返回一个结果对象 CommonResult 的

第 26页 共 149页

JSON 数据,这样如果操作有错误,那么后台可以传递错误信息到界面上,实现友好的提示,这样比简单返回的成功、失败的布尔值有更好的用户体验效果。同时我们在其中预留一个 OnBeforeUpdate 函数供子类进行重写,这样可以在插入前统一修改对象里面的数据。

这个 Insert 函数里面的参数 T,是一个泛型的变量,在子类调用的时候,这个 T 会被替换为具体的实体对象类型,从而得到强类型的接口函数。这样,在视图到控制器 的数据转换中,系统会自动把 Form 里面的集合对象(键值内容)转换为具体的实体类 属性信息,并对应上这个插入的函数。

2.4.2. 界面层的代码

在 index.cshtml 视图代码里面, 里面有一个对话框的 DIV 是用承载增加或编辑的界面内容的。一般通过脚本事件的触发弹出一个添加的界面内容, 在弹出 Div 的过程中对界面进行一定的初始化工作。

```
//弹出新增对话框
function Add() {
    isAddOrEdit = 'add';//新增对话框标识
    //CKEDITOR.instances.Content.setData(''); //清空编辑器的数据
    $("#ffAdd")[0].reset();//清空上次输入
    $('#ffAdd').validate().resetForm();//去除验证信息
    //增加记录前,赋值ID
    url = '/Province/Insert';
    $.get("/Province/NewGuid?r=" + Math.random(), function (result) {
        $("#ID").val(result);
        ID = result;
    });
    //初始化部分控件
    //$('input[name="Gender"][value="男"]').iCheck('check');
    //$("#PID").val(').trigger("change");
    //$("#Hobby").val(null).trigger("change");
    //$("#hobby").val(null).trigger("change");
    $("#add").modal("show");
}
```

一般情况下,我们可能会把增加、编辑界面的 DIV 做成一个,这样方便维护管理,也减少大量相同的界面层代码。

如果是编辑界面,那么我们还涉及到需要把它们的值绑定到控件上显示。



简单的省份信息管理的添加界面效果和代码如下所示。

✔ 添加信息	
省份名称	省份名称
	确定取消
<pre></pre>	本加/修改信息的弹出度 本加/修改信息的弹出度 andal-lg"tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true"> l fade bs-modal-lg"tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true"> g modal-lg"> content"> dal-header bg-primary"> per"button> "modal-title"> ss="con-pencil"> ss="con-pencil"> ss="con-pencil"> ss="icon-pencil">
</td <th><pre></pre></th>	<pre></pre>

对于一般表的字段来说,可能这种展示还是空间太少了,我们可以引入选项卡的方式,

对字段的输入进行归类,也方便用户查看管理。如对于客户的信息管理,界面如下所示。

客户基础信息	客户相关信息 客户状态信息		< → 选项卡进行归	类展示	
客户编号	客户编号		客户名称	客户名称	
客户简称	客户简称		所在省份		
城市		T	所在行政区		
市场分区		~	公司地址	公司地址	
公司邮编	公司邮编		办公电话	办公电话	
传真号码	传真号码		主联系人	主联系人	
联系人电话	联系人电话		联系人手机	联系人手机	
电子邮件	电子邮件		QQ号码	QQ号码	
所属行业		\sim	客户来源		
客户类别		\sim	客户级别		
信用等级	信用等级	\sim	重要级别	重要级别	
公开与否	公开	•	客户阶段	客户阶段	
客户状态	客户状态	~	备注信息	备注信息	

其界面代码如下所示。



在弹出的添加或编辑界面里面,我们还会看到有一个确定的按钮,里面触发的事件就是 我们需要做保存的事件内容了。

```
//绑定相关事件
function BindEvent() {
//添加、编辑记录的窗体处理
   formValidate("ffAdd", function (form) {
    $("#add").modal("hide");
       var postData = $("#ffAdd").serializeArray();
       $.post(url, postData, function (json) {
           var data = $.parseJSON(json);
           if (data.Success) {
               //可增加其他处理
               //保存成功 1.关闭弹出层, 2.刷新表格数据
               showTips("保存成功");
               Refresh();
           }
           else {
               showError("保存失败:" + data ErrorMessage, 3000);
           }
       }).error(function () {
           showTips("您未被授权使用该功能,请联系管理员进行处理。");
       });
   });
```

上面代码里,有一行代码:

```
var postData = $("#ffAdd").serializeArray();
```

就是把整个 Form 里面的元素的值,转换为一个键值集合,然后 POST 提交到指定的控制器 URL 里面了。对于插入记录,它的控制器 URL 类似如下所示,其中 Province 是具体的对象名称:

url = '/Province/Insert';

对于更新的记录,那么它的控制器 URL 就需要变化为如下所示:

url = '/Province/Update?ID=' + ID;

2.5. 删除操作

2.5.1. 控制器后台的操作支持

删除操作也是在控制器基类里面进行处理的,删除操作分为几种,常规的做法是获取界面的删除 ID 集合,然后逐一删除,如下代码所示。

```
/// <summary>
/// 删除多个ID的记录
/// </summary>
/// <param name="ids">多个id组合,逗号分开(1,2,3,4,5)</param>
/// <returns></returns>
2 个引用
public virtual ActionResult DeleteByIds(string ids)
{
   //检查用户是否有权限,否则抛出MyDenyAccessException异常
   base.CheckAuthorized(AuthorizeKey.DeleteKey);
   CommonResult result = new CommonResult();
   try
    {
        if (!string.IsNullOrEmpty(ids))
       {
           List<string> idArray = ids.ToDelimitedList<string>(",");
           foreach (string strId in idArray)
           {
               if (!string.IsNullOrEmpty(strId))
               {
                   baseBLL.Delete(strId);
               }
           }
           result.Success = true;
       }
   }
   catch (Exception ex)
    {
       LogTextHelper.Error(ex);//错误记录
       result.ErrorMessage = ex.Message;
   }
   return ToJsonContent(result);
}
```

还有两种是根据 ID 或者条件进行记录的删除,如下面界面代码所示。

根据 ID 进行记录删除的控制器方法如下所示。

```
/// <summary>
/// 根据指定对象的ID,从数据库中删除指定对象
/// </summary>
/// <param name="id">指定对象的ID</param>
/// <returns>执行成功返回<c>true</c>,否则为<c>false</c>。</returns>
0 个引用
public virtual ActionResult Delete(string id)
{
   //检查用户是否有权限,否则抛出MyDenyAccessException异常
   base.CheckAuthorized(AuthorizeKey.DeleteKey);
   CommonResult result = new CommonResult();
   try
   {
       if (!string.IsNullOrEmpty(id))
       {
           result.Success = baseBLL.Delete(id);
       }
   }
   catch (Exception ex)
    {
       LogTextHelper.Error(ex);//错误记录
       result.ErrorMessage = ex.Message;
   }
   return ToJsonContent(result);
}
```

或者根据条件语句进行记录的删除的控制器方法如下所示。

```
/// <summary>
/// 根据指定条件,从数据库中删除指定对象
/// </summary>
/// <param name="condition">删除记录的条件语句</param>
/// <returns>执行成功返回<c>true</c>,否则为<c>false</c>。</returns>
0 个引用
public virtual ActionResult DeleteByCondition(string condition)
{
    //检查用户是否有权限,否则抛出MyDenyAccessException异常
   base.CheckAuthorized(AuthorizeKey.DeleteKey);
    CommonResult result = new CommonResult();
    try
    {
       if (!string.IsNullOrEmpty(condition))
       {
           result.Success = baseBLL.DeleteByCondition(condition);
       }
    }
    catch (Exception ex)
    Ł
       LogTextHelper.Error(ex);//错误记录
       result.ErrorMessage = ex.Message;
    }
    return ToJsonContent(result);
}
```

以上这些删除操作的控制器方法,都是返回一个通用的结果对象 CommonResult 的 JSON 数据,这种操作是我们常规操作数据库的返回记录,可以承载是否成功,如果有错误 则传递错误等信息,非常方便。

2.5.2. 界面层的代码

在列表界面里面,提供复选框选择要操作的记录,然后操作删除按钮,就会提示确认是 否删除,这个操作界面如下所示。

功能菜单 📃				P 🖉 🖻	👷 管理员 🗸
	▼查询内容		Q 用户测试 Q 测	试 Q查询 团导.	入 【】 导出 →
	省份名称	用户姓名	用户姓名	用户姓名	
会员管理	≪数据列表		+ 新増 🌙	诊改 田 查看 一 删除	そう 日新 23
客户管理	每页显示 10 • 条记	录 共有记录: 📴 条,总页数:	<u>4</u> 页。		
权限管理	□ 编号 図 1	()			操作 Q ≠ ×
系统管理	2 3	操作提示			Q ≠ ¥ Q ≠ ¥
	4 5	您确认删除选定的记录	专吗?		Q / X Q / X
	6	取消 是的,执	行删除!		Q ≠ ¥ Q ≠ ¥
	8	黒龙江省			Q / X

这个效果是引入插件 **sweetalert** (<u>http://lipis.github.io/bootstrap-sweetalert/</u>)实现 的。

```
swal({
    title: "操作提示",
    text: newtips,
    type: "warning", showCancelButton: true,
    confirmButtonColor: "#DD6B55",
    cancelButtonText: "取消",
    confirmButtonText: "是的, 执行删除! ",
    closeOnConfirm: true
```

第 34页 共 149页
```
}, function () {
    delFunction();
});
```

上面的界面效果类似的实现代码如下所示:





一般它的弹出框代码可以做的很简单,如下所示。



而删除操作的脚本代码如下所示。



第 36页 共 149页

其中上面标注的 showDelete 是一个 JS 的公共函数, 其定义在:



里面的完整函数定义如下所示。



2.6. 获取对象

我们需要在界面上显示业务对象的内容,就需要获取业务对象的相关实体类信息,这样 就可以通过脚本绑定到界面控件上。

获取对象,包括获取对象的实体类和实体类集合等操作。

2.6.1. 控制器后台的操作支持

后台控制器基类提供了很多获取对象的操作方法,最常用的是根据 ID 获取对象实体类 类信息,这个实体类包含了对应业务表的字段信息。



获取对象的集合,并返回 JSON 数据的操作如下所示。



2.6.2. 界面层的代码

对于控件绑定对象的属性,我们先要通过 ID 获取到对应的对象(JSON 数据对象),然 后把对象的值赋值给不同的控件,界面代码如下所示。



获取对象并绑定属性到界面上的操作脚本如下所示。



如果是查看界面,则只需要把内容绑定到 Label 控件就可以了。

2.7. 更新操作

更新操作和插入操作类似,都是获取 Form 表单的属性值列表,然后转换为 Json 数据, 提交到具体的控制器更新操作中。在视图里面,通过下面的脚本代码:

```
var postData = $("#ffAdd").serializeArray();
```

把整个 Form 里面的元素的值,转换为一个键值集合,然后 POST 提交到指定的控制器 URL 里面了。

2.7.1. 控制器后台的操作支持



在 Update(id, obj)的操作里面,再次对更新操作进行了一个封装,主要是考虑需要重写的

操作方便,同样更新操作也有一个可供子类实现特殊信息修改的 OnBeforeUpdate 方法。

```
/// <summary>
/// 更新对象属性到数据库中(方便重写的时候操作)
/// </summary>
/// <param name="id">主键ID的值</param>
/// <param name="info">指定的对象</param>
/// <param name="info">指定的对象</param>
/// <returns>执行成功返回<c>true</c> , 否则为<c>false</c>。</returns>
11 个引用
protected virtual bool Update(string id, T info)
{
    OnBeforeUpdate(info);
    return baseBLL.Update(info, id);
}
```

2.7.2. 界面层的代码

```
//修改或查看明细信息(绑定显示数据)
function EditView(view) {
     ID = "";//重置ID的值
$("[name='checkbox']").each(function () {
          if ($(this).is(':checked')) {
               ID = $(this).val();//首先取出来用户选择的数据的ID
          }
     });
     EditViewById(ID, view);
//编辑或者查看指定记录
function EditViewById(ID, view) {
    if (ID == "") {
        showTips("请选择一条记录");
     if (view == null) {
         ///2/24/Big(U)) = 2/
$("#lblAddTitle").text("修改信息");
$("#add").modal("show");
url = '/City/Update?ID=' + ID;
//绑定修改详细信息的方法
          BindEditInfo(ID);
     J
else {
  //处理查看详细
          $("#view").modal("show");
          //绑定查看详细信息方法
          BindViewInfo(ID);
     }
```

更新操作和插入逻辑类似,只是更新操作,它的 url 已经变化为:

url = '/City/Update?ID=' + ID;

更新操作的保存数据脚本如下所示:



2.8. 字典数据绑定

在很多场合,不管是 Winform 还是 Web 的项目,都是要有字典模块的,很多下拉列表 需要绑定字典的值,而且我们也希望在统一的字典管理界面对数据进行维护。而正是由于指 定数据绑定的通用性,我在 CommonUtil.js 脚本里面定义了一个通用的数据字典绑定函数 BindDictItem,如下所示。这样应用了这个脚本的所有视图界面代码,都可以通过这个方法 快速绑定字典类型的值到具体的控件上了。

```
//初始化字典信息(下拉列表)
function InitDictItem() {
    //部分類值参考
BindDictItem("Area","市场分区");
BindDictItem("Industry", "客户行业");
BindDictItem("GustomerType", "客户类型");
BindDictItem("CustomerType", "客户类型");
BindDictItem("CustomerType", "客户类型");
BindDictItem("CreditStatus", "信用等级");
BindDictItem("Status", "ape数别");

//郭定字典
//BindDictItem("Rank", "职务");
//#定树列表
//BindSelect("PID", "/Menu/GetDictIson");
//#定可选多个Tag
// $("#Hobby").select2([
// tags:["旅游","爬山","篮球","足球","乒乓球"]
// });
// #s定省份、城市、行政区(联动处理)
BindSelect("Province", "Province").val();
BindSelect("City", "/City/GetCitysByProvinceNameDictIson?provinceName="+ provinceName", ");
});

$("#City").on("change", function (e) {
    var cityName = $("#City").val();
    BindSelect("District", "/District/GetDistrictByCityNameDictIson?cityName="+ cityName);
});
```

在一般情况下,字典的下拉列表,我们常用到的控件是 Select2,这个控件可以更加丰富传统的 Select 下拉列表控件,提供更多的功能和更好的用户体验。

2.8.1. Select2 控件介绍

这个插件是基于 Select 控件的扩展插件,能够提供更加丰富的功能和用户体验,它的 github 官网地址为:<u>https://select2.github.io/</u>,具体的使用案例,可以参考地址: https://select2.github.io/examples.html。

我们在整个框架里面,用到了很多 Select2 控件来处理内容的显示,包括单选的下拉列表(包括级联选择框)、复选的下拉列表、树形下拉列表等方式,界面效果如下所示。

编辑界面下的省份、城市、所在行政区的级联界面效果,选择省份,会加载对应省份下的城市,选择城市,会继续加载城市下的行政区,从而实现多级关联的下拉列表效果。

基础信息 其他信	息 联系人分组	附件信息 省份	、城市、行政区的级联		
所在省份	□ 广东省		城市) 🗈 广州市	
所在行政区	1	۹	籍贯	广东	
家庭住址	□ 北京市□ 天津市		民族	□ 汉族	×
教育程度	□ 河北省 □ 山西省		毕业学校	西南交通大学	
政治面貌	 □ 内蒙古自治区 □ 辽宁省 		职业类型	☐ 高级管理人员	×
职称	■ 吉林省 ■ 黒龙江省		职务	■ 教授	×
所在部门	开发部		爱好	×旅游 × 爬山 × 篮球	ł
属相	■ 属蛇	× ~	星座	■ 摩羯座	×
婚姻状态	1 已婚	× ~	健康状况	■ 良好	×
体型	〕 偏胖	× ~	吸烟	吸烟	
喝酒	喝酒	•	身高	身高	
体重	体重		视力	视力	
个人简述	个人简述				

2)编辑界面下的多项选择下拉列表

重要级别	■ 非常重要 × ×	认可程度 🖹 忠诚度高 X 🗸
关系	■ 紧密 × ×	标签式的多项选择
负责需求	× 🗈 推广服务 × 📄 维护服务	
关心重点	× 🖹 安全 × 🖹 产品外观形象	
利益诉求	× 🖹 满足需求 × 📄 社会影响	
		确定取消

但我们选择其中的内容的时候,系统自动显示出没有选择的列表数据,非常直观友好, 如下所示。

ш /т	
重要级别	■ 能用
	🖹 价低
关系	☐ 最低价获取
	☐ 提升形象
负责需求	☐ 提升专业成功欲望
	☐ 热心帮助企业成功
关心重点	□ 得到认可
	■ 严格严谨
利益诉求	× 🗈 满足需求 🛛 × 🗈 社会影响

3) 树形列表的下拉列表

有时候,我们的一些数据可能有层次关系的,如所属机构、上层列表等等。

⊘ 修改信息			×
基础信息 用户可	操作功能 所属机构角色 肖像信息		
用户名/登录名*	admin	真实姓名*	管理员
所属公司*	□ 广州份公司 ~	默认部门*	■ 总经办 ^
所属经理	〕无 ∽	职务头衔	♀ ♀ ♀ ♀ ♀ ♀
用户编码	1001	排序码	 ■ 总经办 ■ 财务部
用户呢称	管理员	吸号码	 □ 工程部 □ 产品研发部
邮件地址	wuhuacong@163.com	移动电话	 □ 开发一组 □ 开发二组
身份证号码	身份证号码	性别	■ 测试组
出生日期	2015-08-14	办公电话	办公电话
家庭电话	家庭电话	家庭住址	家庭住址
办公地址	广州市白云区同和路**小区**号		
个性签名	个性签名		
备注	系统管理员		
			确定取消

☎ 修改功能详细信		- 刪除 💱
编号	1	
功能名称*	仓库管理系统 上层功能的树研列表	
上层功能*	■ 企业信息化平台	^
控制标识*	□ 企业信息化亚台	٩
排序码		
系统编号		
	道 登录日志管理	-
	目 月结操作 日 年度汇总操作	

2.8.2. Select2 控件的实际使用代码分析

1) 基础界面代码及操作

使用 select2 控件,一般是在常规的 select 控件上,设置一下即可(设置它的 class

为 select2)。



如果是固定列表,那么也就是设置它的 Option 内容即可,如下所示。



简单的 select2 控件初始化代码如下所示。

```
$(document).ready(function() {
```

\$(".js-example-basic-single").select2();

});

一般情况下,如果允许复选多个项目,那么设置 multiple="multiple"即可,如下代码所示。

<select id="ResponseDemand" name="ResponseDemand" multiple="multiple" class="form-control
select2"></select>

2) 异步数据绑定操作

一般情况下, Select 控件的数据, 是从数据库里面动态加载的, 因此一般是通过 Ajax 方式获取数据并进行绑定即可。

基于代码可重用性的考虑,我们编写一个公用的 **JS** 函数,用来减少绑定操作的代码,提高代码重用性。



这样,绑定公用字典模块的数据,也就可以通过下面进一步封装处理即可。



这样我们初始化 Select2 控件,并动态绑定对应的字典值或者其他数据,则可以通过下面初始化代码即可实现。其中 BindDictItem 就是直接绑定字典内容的操作,BindSelect则是根据 URL 进行数据的获取并绑定,而\$("#Province").on("change", function (e) {});这样的函数处理,就是处理选择内容变化的联动操作了。

<pre>//初始化字典信息(下拉列表) function InitDictItem() //部分赋值参考 BindDictItem("Area","市场分区"); BindDictItem("Industry", "客户行业"); BindDictItem("Grade","客户级别"); BindDictItem("CustomerType", "客户类型"); BindDictItem("Source", "客户来源"); BindDictItem("CreditStatus", "信用等级"); BindDictItem("Status", "客户状态"); BindDictItem("Importance", "重要级别"); //绑定字典</pre>
//绑定树列表 //BindSelect("PID", "/Menu/GetDictJson"); //绑定可选多个Tag
// \$("#Hobby").select2({ // tags: ["旅游", "爬山", "篮球", "足球", "乒乓球"] // });
<pre>// 绑定省份、城市、行政区(联动处理) BindSelect("Province", "/Province/GetAllProvinceNameDictJson"); \$("#Province").on("change", function (e) { var provinceName = \$("#Province").val(); BindSelect("City", "/City/GetCitysByProvinceNameDictJson?provinceName="+ provinceName); });</pre>
<pre>\$("#City").on("change", function (e) { var cityName = \$("#City").val(); BindSelect("District", "/District/GetDistrictByCityNameDictJson?cityName="+ cityName); });</pre>

而其中 MVC 控制器返回的数据,我们是返回一个 JSON 数据列表给前端页面的,他们的数据格式如下所示。

```
[{ "Text": "", "Value": ""}, { "Text": "学术会议", "Value": "学术会议" }, { "Text": "朋友介绍", "Value": "朋友介绍" }, { "Text": "广告媒体", "Value": "广告媒体" } ]
```

这样前端页面绑定 Select2 控件的时候,就使用了 JSON 对象的属性即可。

//绑定Ajax的内容
<pre>\$.getJSON(url, function (data) {</pre>
control.empty();//清空下拉框
<pre>\$.each(data, function (i, item) {</pre>
<pre>control.append("<option value='" + item.Value + "'> " + item.Text + "</option>");</pre>
});
});

2.8.3. Select2 控件的赋值处理

上面介绍的方法,都是介绍 select2 控件的初始化,绑定相关的数据,那么如果初始化 界面后,我们绑定编辑界面的值的时候,就需要赋值给控件,让它显示真正需要显示的项目 了。如清空控件的方法如下所示。 //参考赋值
\$("#Company_ID1").val(info.Company_ID).trigger("change");
\$("#PID1").val(info.PID).trigger("change");//普通Select2
var array = [info.Hobby];
\$("#Hobby").val(array).trigger("change"); //集合

如果对于多个控件, 需要清除, 则可以使用集合进行处理



给 Select2 控件赋值, 让它显示对应值内容的项目, 那么操作也就和上面的类似了。



如果需要级联显示的,那么做法增加一个 onchange 的函数处理就可以了,如下级联 代码的赋值处理如下。

\$("#Province").val(info.Province).trigger('change');//联动
\$("#City").val(info.City).trigger('change');//联动
\$("#District").val(info.District).trigger("change");

2.8.4. 控制器后台的操作支持

控制器的代码只是把字典数据,封装为 CListItem 对象集合,并返回给前端处理即可, 根据字典类型获取集合的控制器的代码如下所示。



2.8.5. 界面层的代码



```
//初始化字典信息
function InitDictItem() {
    BindDictItem("Titles", "职称");
    BindDictItem("Rank", "职务");
    BindDictItem("Importance", "重要级别");
BindDictItem("Recognition", "对公司认可程度");
    BindDictItem("InterestDemand", "客户利益诉求");
    BindDictItem("CareFocus", "客户关心重点");
    BindDictItem("ResponseDemand", "负责需求");
    BindDictItem("RelationShip", "与公司关系");
BindDictItem("Nationality", "民族");
BindDictItem("Political", "政治面貌");
    BindDictItem("JobType", "职业类型");
BindDictItem("Eduction", "学历");
    BindDictItem("Animal", "属相");
    BindDictItem("Constellation", "星座");
    BindDictItem("MarriageStatus", "婚姻状况");
    BindDictItem("HealthCondition", "健康状况");
    BindDictItem("BodyType", "体型");
}
```

2.8.6. 下拉树形列表

对于树形列表,如所属公司、所属部门机构等有层次性的数据,它的绑定操作也是类似的,如下代码所示。



只是它们返回的数据,我们把它作为有缩进的显示内容而已。

```
[{ "Text": "爱奇迪集团", "Value": "1" }, { "Text": " 广州分公司", "Value": "3" }, { "Text": " 上海分公司", "Value": "4" }, { "Text": " 北京分公司", "Value": "5" } ]
```

或者如下所示

```
[{ "Text": "广州分公司", "Value": "3" }, { "Text": "总经办", "Value": "6" }, { "Text": "
财务部", "Value": "7" }, { "Text": "工程部", "Value": "8" }, { "Text": "产品研发部", "Value":
"9" }, { "Text": " 开发一组", "Value": "14" }, { "Text": " 开发二组", "Value": "15" },
{ "Text": " 测试组", "Value": "16" }, { "Text": "市场部", "Value": "10" }, { "Text": "
市场一部", "Value": "23" }, { "Text": " 市场二部", "Value": "24" }, { "Text": "综合部
```

```
", "Value": "11" }, { "Text": "生产部", "Value": "12" }, { "Text": "人力资源部", "Value": "13" } ]
```

而服务后台的控制器代码,就是需要把有层级的数据,转换为有缩进空白字符的 CListItem 集合即可。控制器代码如下所示。



综上所述,我们可以看到它们的 Text 的内容,是根据层次关系进行空格增加,从而实现了层次关系的显示。

黑灯人台灯]*		
职务头衔	□ 爱奇迪集团	
排序码	广州分公司	
	■ 总经办	
QQ号码	■ 财务部	- 1
	□ 工程部	
移动电话	📄 产品研发部	
	日 开发一组	
性别	□ 开发二组	
办公电话	办公电话	

第 54页 共 149页

不过从这个界面效果上讲,这样的处理确实没有 EasyUI 里面,对下拉列表树的展示好 看,也许可以利用更好的 Bootstrap 插件进行这个树形内容的展示。

所属公司(*):	广州分公司	默认部门(*):	总经办
所属经理:	▲ 1 愛奇迪集团 品 广州分公司	职务头衔:	■ 无 ▲
用户编码:	▲ 上海分公司 ▲ 北京分公司	排序码:	当 込み 当 以外部
用户呢称:		QQ号码:	▲ 🗳 工程部 ▲ 🗳 产品研发部
邮件地址:		移动电话:	■
身份证号码:			
性别:		出生日期・	▲市场—部

2.9. 树信息绑定

树形控件的绑定也是在 Web 开发中常见的操作,树形控件显示数据比较直观,因此经常性的用在一些有层次化的数据列表展示里面。

在一般情况下,如果数据有层次的,那么通过树列表展示,可以很直观的显示出它们的 结构,因此树列表在很多情况下,可以辅助我们对数据的分类展示。

如对于用户数据来说,我们可以根据用户的组织机构或者角色进行分类,他们两者可以通过树列表进行直观的展示,这样我们在寻找不同类型的用户列表的时候,就很好找了。

按组织机构查看	按角色查看
○ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	司 办 部 部 部 研 大 次 部 和 大 大 二 组 制 部 市 市 小 二 組 制 部 市 、 、 二 组 制 部 。 本 次 二 组 制 部 。 本 次 二 组 制 部 。 司 司 。 一 、 4 日 一 の 一 の 引 一 の 一 の 司 司 司 司 部 部 部 部 部 部 部 部 部 部 部 部 部 部 。 一 の 日 の 一 の 目 の 一 の 目 の 一 の の 目 の 一 の の の の の

第 55页 共 149页

或者对于字典数据或者省份城市的数据,一样更可以通过树列表进行展示



2.9.1. 插件 JSTree

Bootstrap 的树形控件的绑定,主要引入了 JSTree 的插件,然后通过 Json 数据的传递实 现绑定的。JSTree 控件的官方地址为 <u>https://www.jstree.com/</u>,网站对 JSTree 控件的 使用说明及案例讲解的已经很清晰了,一般情况下,我们直接参考例子就可以使用了。 简单的 JSTree 使用代码如下所示

```
$(function () { $('#jstree_demo_div').jstree(); });
```

对于 JSTree 的事件,我们一般可以通过下面代码进行绑定事件。

\$('#jstree_demo_div').on("changed.jstree", function (e, data) {

第 56页 共 149页

```
console.log(data.selected);
```

});

JSTree 一般我们会通过 JSON 数据进行动态绑定,这个 JSON 的数据格式定义如下所示。

```
{
           : "string" // required
 id
           : "string" // required
 parent
            : "string" // node text
 text
            : "string" // string for custom
 icon
           : {
 state
            : boolean // is the node open
   opened
   disabled : boolean // is the node disabled
   selected : boolean // is the node selected
 },
 li attr
           : {} // attributes for the generated LI node
 a attr : {} // attributes for the generated A node
}
```

根据上面 JSON 的定义,我们通过指定一个 JsTreeData 对象的 JSON 集合,即可赋值 给对应的树控件了。JsTreeData 的对象,主要用来承载 JSTree 的 JSON 对象信息,实体 类代码如下所示。

```
/// <summar
/// JsTree的数据模型
/// </summary
[Serializable]
public class JsTreeData
{
    /// <summary>
    /// 默认构造函数
   /// </summary>
30 个引用
   public JsTreeData()...
   /// <summary> ...
   public JsTreeData(object id, string text, string icon = null) : this()...
   /// <summary> ...
   [DataMember]
5 个引用
    public string id { get; set; }
    /// <summarv>
    /// 节点显示名称
    /// </summary>
   [DataMember]
     人司田
   public string text { get; set; }
    /// <summary>
    /// 图标
    /// </summary>
    [DataMember]
   [JsonProperty(NullValueHandling = NullValueHandling.Ignore)]
2 个引用
    public string icon { get; set; }
   /// <summary>
   /// 状态对象
    /// </summary>
    [DataMember]
   [JsonProperty(NullValueHandling = NullValueHandling.Ignore, DefaultValueHandling = DefaultValueHandling.Ignore)]
    3 个引用
   public JsTreeState state { get; set; }
   /// <summary>
/// 子节点集合
    /// </summary>
    [DataMember]
    [JsonProperty(NullValueHandling = NullValueHandling.Ignore, DefaultValueHandling = DefaultValueHandling.Ignore)]
      个引用
    public List<JsTreeData> children { get; set; }
}
```

一般情况下,我们通过下面的脚本进行数据的清空和绑定操作

```
$('#jstree_demo_div').data('jstree', false);//清空数据,必须
//异步进行JSON数据的绑定
$.getJSON(url, function (data) {
    $('#jstree_demo_div').jstree({
        'core': {
            'data': data,
            "themes": {
                 "responsive": false
            }
        }).bind('loaded.jstree', loadedfunction);
});
```

2.9.2. JSTree 插件的复选框处理

如果我们需要给用户提供复选框,设置 JSTree 的选中状态,界面效果如下所示。

📽 修改角色详细信息	+ 新増 - 刪除 🔰
角色详细信息 可操作功能 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	可访问数据 JSTree带复选框,不级联选择
	保存

那么一般的初始化函数就需要变化一下,如下代码所示

```
//带复选框的JSTree的初始化代码
$.getJSON(url, function (data) {
    control.jstree({
        'plugins': [ "checkbox" ], //出现选择框
        'checkbox': { cascade: "", three_state: false }, //不级联
        'core': {
            'data': data,
            "themes": {
                "responsive": false
            }
        }).bind('loaded.jstree', loadedfunction);
});
```

综合两者,我们可以进一步把 JSTree 控件的初始化绑定提炼为一个 JS 的公共函数 bindJsTree,放在 CommonUtil.js 文件里面实现共用的 JS 函数即可。

```
//treeName为树div名称, url为数据源地址, checkbox为是否显示复选框, loadedfunction为加载完毕的回调函数
function bindJsTree(treeName, url, checkbox, loadedfunction) {
    var control = $('#' + treeName)
    control.data('jstree', false);//清空数据,必须
     var isCheck = arguments[2] || false; //设置checkbox默认值为false
     if(isCheck) {
           //发达性的10000476
$.getJSON(url, function (data) {
    control.jstree({
        'plugins':["checkbox"], //出现选择框
        'checkbox': { cascade: "", three_state: false }, //不级联
    }
}
                        'core': {
____data': data,
                             "themes": {
                                    "responsive": false
                 }).bind('loaded.jstree', loadedfunction);
     }
else {
    //普通树列表的初始化
           $.getJSON(url, function (data) {
                 control.jstree({
                        'core': {
'data': data,
                             "themes": {
                                   "responsive": false
                             }
                 }).bind('loaded.jstree', loadedfunction);
```

因此在页面的绑定 JSTree 的时候,代码可以简化一下。

```
//初始化组织机构列表
function initDeptTreeview() {
    var treeUrl = '/User/GetMyDeptJsTreeJson?userId=@Session["UserId"]';
    bindJsTree("jstree_div", treeUrl);
    //树控件的变化事件处理
    $('#jstree_div').on("changed.jstree", function (e, data) {
        var icon = data.node.icon;
        loadDataByOu(data.selected);
    });
}
```

对于复选框的列表,初始化代码如下所示。

```
//初始化所有该用户的所有功能集合
var treeUrl = '/Function/GetRoleFunctionJsTreeByUser?userId=@Session["UserId"]';
bindJsTree("tree_function", treeUrl, true);
//角色数据权限,先初始化所有部门
treeUrl = '/User/GetMyDeptJsTreeJson?userId=@Session["UserId"]';
```

```
bindJsTree("tree roledata", treeUrl, true);
```

对于复选框,我们一般是初始化数据,然后在根据需要设置树列表的选中状态,这种不 用频繁初始化树,可以有效提高性能和响应体验。

那么我们在初始化树列表后,就需要清空选择项,然后设置我们所需要的选择项即可, 具体代码如下所示,注意其中的 uncheck all 和 check node 事件的处理。

```
//初始化角色数据权限集合(组织机构)
function initRoleData(id) {
    if (id != "") {
        var treeMenu = "tree_roledata";
        $('f' + treeMenu).jstree("uncheck_all");//取消所有选中
        //勾选指定内容
        $.getJSON("/RoleData/GetRoleDataList?r=" + Math.random() + "&roleId=" + id, function (json) {
            $.each(json, function (i, item) {
                $('f' + treeMenu).jstree('check_node', item);//将节点选中
            });
        });
    }
}
```

数据保存的时候,我们获得 JSTree 的节点选中列表就可以进行数据的保存了,具体代码如下所示。

```
//保存角色数据权限
function saveRoleData(roleid) {
    var ouList = $('#tree_roledata').jstree('get_selected');
    var url = '/RoleData/UpdateData?r=' + Math.random();
    var postData = { roleId: roleid, ouList: ouList.join(',')};
    $.post(url, postData, function (json) {
        initRoleData(roleid);
    }).error(function () {
        showTips("您未被授权使用该功能,请联系管理员进行处理。");
    });
}
```

界面效果如下所示。

角色信息				
▼ 角色列表	Β ≒ λ	日常出	C Risi	∞ 修改角色详细信息
 ● 愛奇迪集団 ● 超級管理员 ● 一番系统管理员 ● 二番 广州分公司 ● 急经理 ● 急援管理 ● 最終管理 ● 最終管理 ● 御行経理 				 角色详细信息 可操作功能 可读问数据 ○ 企址信息化平台 ○ 公 令年管理系统 ○ 公 零出出库単 ○ 公 零出出库単 ○ 公 零出出库単 ○ 公 零出入库単 ○ 公 零出入库申 ○ 公 零計算用表 ○ 公 零計算用表 ○ 公 零計算用表 ○ 公 零 ○ 公 零 ○ 公 報告 ○ 公 電子 ○ 公 報告 ○ 公 零 ○ 公 報告 ○ 公 報告 ○ 公 零 ○ 公 電話 ○ ○ ○ 電話 ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

2.9.3. 控制器后台的操作支持

前面介绍了根据 JSTree 插件的数据绑定 JSON 的定义,我们通过指定一个 JsTreeData 对象的 JSON 集合,即可赋值给对应的树控件。

/// <su< th=""><th>immary></th></su<>	immary>
/// 获明	双用户的可操作功能
/// <td>summary></td>	summary>
/// <pa< td=""><td>aram name="userId"></td></pa<>	aram name="userId">
/// <re< td=""><td>eturns></td></re<>	eturns>
0 个引用	
public	ActionResult GetrunctionJsireeJsonbyUser(int useria)
۱ Lis	<pre>st<]sTreeData> treeList = new List<]sTreeData>();</pre>
Lis for	<pre>st<systemtypeinfo> typeList = BLLFactory<systemtype>.Instance.GetAll(); reach (SystemTypeInfo typeInfo in typeList)</systemtype></systemtypeinfo></pre>
ĩ	<pre>JSTreeData parentNode = new JsTreeData(typeInfo.OID, typeInfo.Name, "fa fa-sitemap icon-state-warning icon-lg"); List<functionnodeinfo> list = BLLFactory<function>.Instance.GetFunctionNodesByUser(userId, typeInfo.OID); AddJsTreeeFunctionNode(parentNode, list);</function></functionnodeinfo></pre>
	treeList.Add(parentNode):
}	
if	(treeList.Count == 0)
{	
	treeList.Insert(0, new JsTreeData(-1, "无"));
}	
ret }	turn ToJsonContent(treeList); 递归方法
3 个引用	
private	e void AddJsTreeeFunctionNode(JsTreeData node, List <functionnodeinfo> list)</functionnodeinfo>
۱ for	peach (EurotionNedeInfe infe in list)
{	each (Functionmodelino ino in 115c)
t	<pre>JsTreeData subNode = new JsTreeData(info.ID, info.Name, info.Children.Count > 0 ? "fa fa-users icon-state-info icon node.children.Add(subNode);</pre>
	AddlsTreeeFunctionNode(subNode, info.Children):
}	
}	

或类似的其他操作方法。

```
/// <summary>
/// 获取用户的公司结构(分级需要) (bootstrap的JSTree)
/// </summary>
/// <param name="userId">用户ID</param>
/// <returns></returns>
0 个引用
public ActionResult GetMyCompanyJsTreeJson(int userId)
ſ
   List<JsTreeData> treeList = new List<JsTreeData>();
   UserInfo userInfo = BLLFactory<User>.Instance.FindByID(userId);
    if (userInfo != null)
    {
        List<OUNodeInfo> list = new List<OUNodeInfo>();
       if (BLLFactory<User>.Instance.UserInRole(userInfo.Name, RoleInfo.SuperAdminName))
        {
           list = BLLFactory<OU>.Instance.GetGroupCompanyTree();
       }
        else
        {
           OUInfo myCompanyInfo = BLLFactory<OU>.Instance.FindByID(userInfo.Company_ID);
           if (myCompanyInfo != null)
            {
               list.Add(new OUNodeInfo(myCompanyInfo));
            }
        }
       if (list.Count > 0)
        {
           OUNodeInfo info = list[0];//无论是集团还是公司,节点只有一个
            JsTreeData node = new JsTreeData(info.ID, info.Name, GetBootstrapIcon(info.Category));
           GetJsTreeDataWithOUNode(info.Children, node);
           treeList.Add(node);
       }
   }
    return ToJsonContent(treeList);
```

因此,一般的树形列表数据,基本上都通过递归的方式,把树形列表的数据放到 JSTreeData 对象集合里面,然后提供给前端页面调用。

2.9.4. 界面层的代码

标准的 JSTree 插件绑定界面如下所示。

var treeUrl = '/Function/GetFunctionJsTreeJsonByUser?userId=' + info.ID; bindJsTree("jstree_function", treeUrl);

如果需要处理树形控件的响应操作,那么处理它的"changed.jstree"事件即可,如下例子代码所示。



如果需要带有复选框的选择框的,那么在 bindJSTree 的函数后面,增加一个布尔变量, 一般为 True 就可以显示为可选择的界面了。



第 65页 共 149页



或者设置的时候,也可以通过 JS 来绑定树形列表的选择那些值,具体代码例子如下所示。



2.10. 文件上传插件 File Input 的使用

Bootstrap 文件上传插件 File Input 是一个不错的文件上传控件,但是搜索使用到的 案例不多,使用的时候,也是一步一个脚印一样摸着石头过河,这个控件在界面呈现上,比 使用过的 Uploadify 好看一些,功能也强大些。

2.10.1. 文件上传插件 File Input 介绍

这个插件主页地址是: <u>http://plugins.krajee.com/file-input</u>,可以从这里看到很多 Demo 的代码展示: <u>http://plugins.krajee.com/file-basic-usage-demo</u>。

这是一个增强的 HTML5 文件输入控件,是一个 Bootstrap 3.x 的扩展,实现文件上传预览,多文件上传等功能。一般情况下,我们需要引入下面两个文件,插件才能正常使用:

bootstrap-fileinput/css/fileinput.min.css

bootstrap-fileinput/js/fileinput.min.js

简单的界面效果如下所示,和众多上传文件控件一样,可以接受各种类型的文件。当然, 我们也可以指定具体接受的文件类型等功能。

Select File					
	🗁 Browse				
<label class="control-label">Select File</label> <input_id="input_1"_type="file" class="file"></input_id="input_1"_type="file">					
	<pre>:t File <label class="control-label">Select File</label> <input class="file" id="input-1" type="file"/></pre>				

如果需要考虑中文化,那么还需要引入文件:

bootstrap-fileinput/js/fileinput_locale_zh.js

这样基于 MVC 的 Bundles 集合,我们把它们所需要的文件加入到集合里面即可。

//添加对 bootstrap-fileinput 控件的支持

css_metronic. Include("~/Content/MyPlugins/bootstrap-fileinput/css/fileinput.min.css");

js_metronic.Include("~/Content/MyPlugins/bootstrap-fileinput/js/fileinput.min.js");

js_metronic.Include("^/Content/MyPlugins/bootstrap-fileinput/js/fileinput_locale_zh.js");

这样我们在页面里面,就可以呈现出中文的界面说明和提示了,如下界面所示。

✔ 修改信息			×
姓名 *	李四	手机 *	1359888887
邮箱 *	wuhuacong@gmail.com	主页	http://wwww.iqidi.com
兴趣爱好	×旅游 × 篮球 × 乒乓球	性别	○ 男 🖌 女
		年龄	- 25 +
出生日期	1999-01-31	身高	- 185 +
备注	同事		
肖像	 ○ 选择 ··· 		×
			确定取消

2.10.2. 文件上传插件 File Input 的使用

一般情况下,我们可以定义一个 JS 的通用函数,用来初始化这个插件控件的,如下 JS 的函数代码所示。

```
//初始化 fileinput 控件(第一次初始化)
function initFileInput(ctrlName, uploadUrl) {
  var control = $('#' + ctrlName);
  control.fileinput({
    language: 'zh', //设置语言
    uploadUrl: uploadUrl, //上传的地址
    allowedFileExtensions : ['jpg', 'png','gif'],//接收的文件后缀
    showUpload: false, //是否显示上传按钮
    showCaption: false,//是否显示标题
    browseClass: "btn btn-primary", //按钮样式
    previewFileIcon: "<i class='glyphicon glyphicon-king'></i>",
});
```

}

页面代码里面,我们放置一个文件上传控件,如下代码所示。

</div>

这样我们脚本代码的初始化代码如下:

```
//初始化 fileinput 控件(第一次初始化)
initFileInput("file-Portrait", "/User/EditPortrait");
```

这样就完成了控件的初始化了,如果我们需要上传文件,那么还需要 JS 的代码处理客户

端上传的事件,同时也需要 MVC 后台控制器处理文件的保存操作。

例如我对窗体数据的保存处理代码如下所示。



其中我们注意到文件保存的处理逻辑代码部分:

```
//增加肖像的上传处理
initPortrait(data.Datal);//使用写入的 ID 进行更新
$('#file-Portrait').fileinput('upload');
```

第一行代码就是重新构建上传的附加内容,如用户的 ID 信息等,这样我们就可以根据

这些 ID 来构建一些额外的数据给后台上传处理了。

这个函数主要就是重新给 ID 赋值,方便上传的时候,获取最新的附加参数,这个和 Uploadify 的处理模式一样的。

前面我们看到,上传的地址为: "/User/EditPortrait",这个后台的函数如下所示。



这样我们就构建了上面的用户肖像的保存处理逻辑了,文件可以正常的保存到后台的文件系统里面,同时数据库里面记录一些必备的信息。

当然,除了用来处理用户的肖像图片,我们也可以用来构建图片相册的处理操作的,具体界面如下所示。




2.10.3. bootstrap-fileinpu 实现数据导入处理

在前面内容中,介绍了 Bootstrap-table 表格插件的在排序分页中的应用过程,这里 介绍另外一个 Bootstrap FieInput 插件的使用,整合两者可以实现我们常规的 Web 数据 第 71页 共 149页 导入操作,导入数据操作过程包括有上传文件,预览数据,选择并提交记录等一系列操作。

在页面中,我们使用以下 HTML 代码实现界面展示,主要的 bootstrap fileinput 插件声明, 主要是基础的界面代码

<input id="excelFile" type="file">

Excel 导入的的界面展示如下所示。

文件导入	
	IB TestUser-模板.xls
Excel导入操作	/ 출选择
	关闭保存

选择指定文件后,我们可以看到 Excel 的文件列表,如下界面所示。

文件导入	
	IB TestUser-模板.xls
TestUser-摸板.xls	×
TestUserxls	直移除 ④上传 广选译…
	关闭保存

上传文件后,数据直接展示在弹出层的列表里面,这里直接使用了 Bootstrap-table 表格 插件进行展示。

文件导入										
	■ TestUser-模板.xls									
	► y4-y2									
									S	₩
	姓名	手机	邮箱	±页	兴趣爱好	性别	年龄	出生日期	身高	备注
	张三	18620292076	wuhuacong@163.com	http://www.iqidi.com	旅游,篮 球,乒乓 球	男	30	1995- 01-01	175	朋友
	李四	1359888887	wuhuacong@gmail.com	http://www.iqidi.com	旅游,篮 球,乒乓 球	男	25	1999- 01-31	185	同事
4										
广州爱奇迪 伍华聪 http://www.iqidi.com										

这样我们就可以把 Excel 的记录展示出来,实现了预览的功能,勾选必要的记录,然 后保存即可提交到服务器进行保存,实现了 Excel 数据的真正导入数据库处理。

我们在实际导入 Excel 的界面中,HTML 代码如下所示。



对于 bootstrap fileinput 的各种属性,我们这里使用 JS 进行初始化,这样方便统一管理

和修改。

```
//初始化Excel导入的文件
function InitExcelFile() {
   //记录GUID
   $("#AttachGUID").val(newGuid());
   $("#excelFile").fileinput({
       uploadUrl: "/FileUpload/Upload",//上传的地址
                                    //异步上传
       uploadAsync: true,
                                    //设置语言
       language: "zh",
       showCaption: true,
                                    //是否显示标题
                                    //是否显示上传按钮
       showUpload: true,
                                    //是否显示移除按钮
       showRemove: true,
                                    //是否显示预览按钮
       showPreview : true,
       browseClass: "btn btn-primary", //按钮样式
       dropZoneEnabled: false,
                                    //是否显示拖拽区域
       allowedFileExtensions: ["xls", "xlsx"], //接收的文件后缀
                                            //最大上传文件数限制
       maxFileCount: 1,
       previewFileIcon: '<i class="glyphicon glyphicon-file"></i>',
       allowedPreviewTypes: null,
       previewFileIconSettings: {
           'docx': '<i class="glyphicon glyphicon-file"></i>',
           'xlsx': '<i class="glyphicon glyphicon-file"></i>',
           'pptx': '<i class="glyphicon glyphicon-file"></i>',
           'jpg': '<i class="glyphicon glyphicon-picture"></i>',
           'pdf': '<i class="glyphicon glyphicon-file">>/i>',
           'zip': '<i class="glyphicon glyphicon-file">>/i>',
       },
       uploadExtraData: { //上传的时候, 增加的附加参数
          folder: '数据导入文件', guid: $("#AttachGUID").val()
       - }-
   }) //文件上传完成后的事件
```

```
}) //文件上传完成后的事件
.on('fileuploaded', function (event, data, previewId, index) {
   var form = data.form, files = data.files, extra = data.extra,
       response = data.response, reader = data.reader;
   var res = data.response; //返回结果
   if (res.Success) {
       showTips('上传成功');
       var guid = $("#AttachGUID").val();
       //提示用户Excel格式是否正常,如果正常加载数据
       $.ajax({
          url: '/TestUser/CheckExcelColumns?guid=' + guid,
          type: 'get',
          dataType: 'json',
          success: function (data) {
              if (data.Success) {
                  InitImport(guid); //重新刷新表格数据
                  showToast("文件已上传,数据加载完毕!");
                  //重新刷新GUID,以及清空文件,方便下一次处理
                 RefreshExcel():
              3
              else {
                  showToast("上传的Excel文件检查不通过。请根据页面右上角的Excel模板格式进行数据录入。", "error");
              3
          }
       });
   }
   else {
       showTips('上传失败');
```

上面的逻辑具体就是,设置上传文件的后台页面为:/FileUpload/Upload,以及各种插件的配置参数,uploadExtraData 里面设置的是提交的附加参数,也就是后台控制器接收的参数,其中

.on('fileuploaded', function (event, data, previewId, index) {

的函数处理文件上传后的处理函数,如果上传文件返回的结果是成功的,那么我们再次调用 ajax 来检查这个 Excel 的字段是否符合要求,如下地址:

url: '/TestUser/CheckExcelColumns?guid=' + guid,

如果这个检查的后台返回成功的记录,那么再次需要把 Excel 记录提取出来预览,并清空 bootstrap fileinput 文件上传插件,方便下次上传文件。如下代码所示。



其中 RefreshExcel 就是重新更新上传的附加参数值,方便下次上传,否则附加参数的值一

直不变化,就会导致我们设置的 GUID 没有变化而出现问题。

```
//重新更新GUID的值,并清空文件
function RefreshExcel() {
    $("#AttachGUID").val(newGuid());
    $('#excelFile').fileinput('clear');//清空所有文件
    //附加参数初始化后一直不会变化,如果需要发生变化,则需要使用refresh进行更新
    $('#excelFile').fileinput('refresh', {
        uploadExtraData: { folder: '数据导入文件', guid: $("#AttachGUID").val() },
    });
}
```

而其中 InitImport 就是获取预览数据并展示在 Bootstrap-table 表格插件上的。



最后就是确认提交后,会通过 JS 提交数据到后台进行处理,如下代码所示。

```
//保存导入的数据
function SaveImport() {
   var list = [];//构造集合对象
   var rows = $import.bootstrapTable('getSelections');
   for (var i = 0; i < rows.length; i++) {</pre>
       list.push({
          'Name': rows[i].Name, 'Mobile': rows[i].Mobile, 'Email': rows[i].Email, 'Homepage': rows[i].Homepage,
          'Hobby': rows[i].Hobby, 'Gender': rows[i].Gender, 'Age': rows[i].Age, 'BirthDate': rows[i].BirthDate,
          'Height': rows[i].Height, 'Note': rows[i].Note
      });
  }
   if (list.length == 0) {
       showToast("请选择一条记录", "warning");
       return;
   }
   var postData = { 'list': list };//可以增加其他参数,如{ 'list': list, 'Rucanghao': $("#Rucanghao").val() };
   postData = JSON.stringify(postData);
   $.ajax({
      url: '/TestUser/SaveExcelData',
      type: 'post',
      dataType: 'json',
      contentType: 'application/json;charset=utf-8',
       traditional: true,
       success: function (data) {
          if (data.Success) {
              //保存成功 1.关闭弹出层,2.清空记录显示 3.刷新主列表
              showToast ("保存成功");
```

这里我们的 JS 代码里面,涉及了几个 MVC 后台的方法处理: Upload、 CheckExcelColumns、GetExcelData、SaveExcelData。这里分别进行介绍。

文件上传的后台控制器方法如下所示。

```
/// <summary>
/// 上传附件到服务器上
/// </summary>
/// <param name="fileData">附件信息</param>
/// <param name="guid">附件组GUID</param>
/// <param name="folder">指定的上传目录</param>
/// <returns></returns>
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Upload(string guid, string folder)
{
    CommonResult result = new CommonResult();
   HttpFileCollectionBase files = HttpContext.Request.Files;
    if (files != null)
    {
       foreach (string key in files.Keys)
        {
            try
            {
                #region MyRegion
                HttpPostedFileBase fileData = files[key];
                if (fileData != null)
```

文件上传处理后,返回一个通用的 CommonResult 的结果对象,也方便我们在 JS 客户端 进行判断处理。

而其中检查我们导入 Excel 的数据是否满足列要求的处理,就是判断它的数据列和我们预 先设置好的列名是否一致即可。

```
//导入或导出的字段列表
string columnString = "姓名,手机,邮箱,主页,兴趣爱好,性别,年龄,出生日期,身高,备注";
/// <summary>
/// 检查Excel文件的字段是否包含了必须的字段
/// </summary>
/// <param name="guid">附件的GUID</param>
/// <returns></returns>
public ActionResult CheckExcelColumns(string guid)
-{
   CommonResult result = new CommonResult();
   trv
   {
       DataTable dt = ConvertExcelFileToTable(guid);
      if (dt != null)
          //检查列表是否包含必须的字段
          result.Success = DataTableHelper.ContainAllColumns(dt, columnString);
       3
```

而 GetExcelData 则是格式化 Excel 数据到具体的 List < TestUserInfo > 集合里面,这样 我们方便在客户端进行各种属性的操作,它的代码如下所示。

```
/// <summary>
/// 获取服务器上的Excel文件,并把它转换为实体列表返回给客户端
/// </summary>
/// <param name="guid">附件的GUID</param>
/// <returns></returns>
public ActionResult GetExcelData(string guid)
₹.
    if (string.IsNullOrEmpty(guid))
    {
       return null;
    3
    List<TestUserInfo> list = new List<TestUserInfo>();
    DataTable table = ConvertExcelFileToTable(guid);
    if (table != null)
    Ł
       #region 数据转换
       int i = 1;
       foreach (DataRow dr in table.Rows)
        {
```

上面这几个函数的代码一般是比较有规律的,不需要一个个去编写,一般通过代码生成工具 Database2Sharp 批量生成即可。这样可以有效提高 Web 的界面代码和后台代码的开发 效率,减少出错的机会。 整个导入 Excel 数据的处理过程,所有代码都贴出来了,基本上整个逻辑了解了就可以很好的了解这个过程的代码了。

2.11. 文件上传 Uploadify 的使用(停用)

(这个插件由于浏览器支持问题,已经使用更好的 Bootstrap FieInput 插件替代生成)

在很多场合,我们需要通过 Web 方式从客户端上传一些文件到服务器端,而在这方面, Uploadify 这个控件做的非常不错。

Uploadify 是 JQuery 一个著名的上传插件,利用 Flash 技术,Uploadify 越过浏 览器的限制,控制了整个上传的处理过程,实现了客户端无刷新的文件上传,这样就实现了 在客户端的上传进度控制,所以,你首先要确定浏览器中已经安装了 Adobe 的 Flash 插 件。Uploadify 当前有两个版本,基于 Flash 是免费的,还有基于 HTML5 的收费版, 我们这里使用比较广泛的免费版。

这个组件需要 Jquery 库的支持,一般情况下,需要添加 Jquery 的 js 库,如下所示: <script type="text/javascript" src="~/Scripts/jquery-2.0.3.min.js"></script>

不过由于我的 Web 开发框架是基于 EasyUI 的,一般在网页的开始就会引用相关的类库, 已经包含了 Jquery 的类库了,所以上面的不需要独立引用。一般情况下,我们只需要添加 Javascript 类库(jquery.uploadify.js),另外加上他的样式文件(uploadify.css)即可, 不过我们在 MVC 里面,已经利用了 BundleConfig 引入了对应的 JS 和 CSS 文件,不用 再操心了。

为了实现文件的上传,一般需要在 HTML 代码中放置两个控件,一个是用来上传的控件, 一个是用来显示已上传列表的控件,还有就是添加上传和取消上传的按钮操作,如下所示。

修改信息 原系人基本信息 アメリア アメージョン	湘关信息		
浏览 ■ 上传 ¥ 取消	1	文件上传部分	
附件暂无	2	文件列表部分	

第 81页 共 149页

如果是编辑状态下,那么列表就是实际已经上传的文件了。

●修改信息
IIII 联系人基本信息 // 其他相关信息 // 附件信息
浏览 ■ 上传 ¥ 取消 1
 [附件1] ● 职位.xls [附件2] ● 开发框架-系统功能介绍白皮书.doc [附件3] ● DotNET共享软件制作经验分享.ppt [附件4] ● uploadify.zip [附件5] ● offer.docx [附件6] ● AllCityData.rar [附件7] ● company.jpg [附件8] ● 企业级管理软件快速平台.ppt 2 [附件9] ● 合并逻辑.doc [附件10] ● 需求.xls [附件11] ● web1002.png [附件12] ● web4025.png
● 全部打包加载

2.11.1. 控制器后台的操作支持

所有的附件上传操作,都是由控制器 FileUploadController 进行处理,把它存储在 对应的 TB_FileUpload 表里面,这个和 Winform 框架的附件存储方式保持一致,这样所 有的附件都是存储在这个表里面。这个附件管理的控制器 FileUploadController 主要管 理附件的上传、删除、下载、列表界面生成、查看附件代码等操作功能。

文件名称: Web 框架界面代码分析



控制器中,对附件上传的操作代码逻辑,主要就是解析传入的参数和里面的

HttpPostedFileBase 文件集合, 然后存储到后台的数据库即可。

<pre>FileUploadInfo info = new FileUploadInfo(); info.FileData = ReadFileBytes(fileData); if (info.FileData != null)</pre>
{
info.FileSize = info.FileData.Length
}
info.Category = folder;
info.FileName = fileName;
info.FileExtend = fileExtension;
info.AttachmentGUID = guid;
info.AddTime = DateTime.Now;
info.Editor = CurrentUser.Name;//登录人
//info.Owner_ID = OwerId;//所属主表记录ID
<pre>result = BLLFactory<fileupload>.Instance.Upload(info);</fileupload></pre>

2.11.2. 界面层的代码

界面代码如下所示。



对界面进行初始化的脚本函数代码如下所示。



在上面的脚本中,均有注释,一看就可以明白相关的属性了,不明白的也可以到官方网站去查找了解。值得注意的就是:

'uploader': '/FileUpload/Upload'

这行就是提交文件给 MVC 的 Action 进行处理,我们在控制器 FileUpload 的 Upload 处理即可。

另外,在附件上传完毕后,我们需要对所在的界面进行更新,以便显示已上传的列表, 那么我们需要增加下面的函数处理即可。

'onQueueComplete': function (event, data) {

最后说明非常值得注意的地方,就是文件上传的时候,我们需要动态获取界面上的一些

元素的值,作为参数传递,那么我们就需要在 on UploadStart 函数中进行如下处理。

\$("#file_upload").uploadify("settings", 'formData', { 'folder': '政策法规', 'guid': \$("#Attachment_GUID").val() }); //动态传参数

为了操作方面,我们一般把一些脚本逻辑封装在常用的 JS 文件 CommonUtil.js 里面,这 样我们一个可以实现通用目的,二也有准确的智能提示,非常方便。

第 84页 共 149页

上面操作在文件上传完成后,马上会调用 ShowUpFiles 函数更新附件列表的信息,绑定到 控件层上,这样我们就能及时看到文件的更新列表了。



2.12. 附件管理和图片预览、Office 文档预览

在上面的附件上传到后台后,一般情况下,需要删除、下载、查看详细的文档内容等操 作,对于一般的文件我们可能提供下载就可以了,对于图片,我们可以实现在线的图片预览 功能。

下面是几种格式文件的查看效果。

Excel 文件预览效果如下所示。

查	看文件								
查: 序 1 2 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 8 9 10 11 12 13 14 5 16 7 18 9 10 11 12 13 14 15 16 17 18 19 10 10 10 10 10 10 10 10 10 10	着文件 引力に編載 引力に編載 引力に編載 引力に編載 引力に編載 引うしたい。 引うしたい	3用户名/受売 shadmin bjadmin gzadmin 100001 100003 100004 100006 100009 100012 100015 100015 100015 100021 100022 100023 100023 100024 100025 100028	,一个陈阮邓黄王朱王杨包黄周邱张陈黄郭韩徐有京,州莉琪卉晓景丽伟维英志黎海烨晓乐锦冬莉丝管理理理理,一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一	头衔 移动电话 13810107123 13575509680 13917028898 18674042588 13761601760 13661246623 13370717219 13996351507 18628056055 18696665275 13452332225 13102321127 18623555060 18623400401 18602333677 13883844451 13594136753 18623490678	办公电话	邮件地址	<mark>性</mark> 男男男男女女男男男男女男男女女男男男女女男男男男女女男男男男女女男男男女女	5 上海分公司管理员 北京分公司管理员 广州分公司管理员	×
21 22 23 24	100035 100036 100037 100039	100036 100037 100039	^{环利} 王海燕 高妮妮 张乐君	13824779380 13616512799 13906696187			_另 男 男		

第 85页 共 149页

word 文档预览效果如下所示:

查看文件
录用通知书
尊敬的周婷婷小姐:
我代表上海金略软件技术有限公司很高兴地通知您,您已经通过了公司的面试考核,公司拟录用您为正式员工并拟与您签订正式劳动合 同。欢迎您加入公司销售部,暂任销售代表职位。
您入职后的薪酬待遇:
您转正后的税前月基本工资是人民币6000元整,全年12个月薪资,随月度工作情况浮动发放,试用期底薪是人民币5000元整,试用期3 月,业绩提成10%。公司将从您的月工资中按国家劳动法规定代扣您个人所得税的个人缴纳部分。
您入职后的福利待遇:
社会福利: 在入职后的6个月开始, 按照国家及地方政府规定缴纳的养老保险、失业保险、基本医疗保险、住房公积金, 公司将从您的人 工资中代扣个人应缴部分;
您入职后的年度薪酬调整:
公司每年将结合市场薪酬水平、岗位调整、个人绩效考核的结果进行薪酬调整。
本通知书的确认及报到
请您于 2014年5月 1日前回夏确认接受此录用通知书并于2014年6月1日前携带已签字确认的录用通知书到公司报到。
第一天入职时,您需要携带以下文件:
身份证(原件及复印件)
毕业证和学位证(原件及复印件)
一寸照片(2 张): 近期免冠白底彩色照片
医疗蓝本或社保卡
逾期未确认接受此录用通知书或逾期未报到,以及报到时不能提供上述真实有效文件的,本通知书自始不发生法律效力。
其他说明:
1.工作时间:公司的工作时间为早晨9:00至下午12:00。中午13:30至: 为 午餐及休息时间。第一天入职时间是上午 9:

或者我们查看图片文件的时候,可以获得界面效果如下所示:

基础信息 其他信息 联系人分组 附件信息	
浏览 上传 取消	
 ו[附件1] 3. jpg ו[附件2] test. jpg ו[附件3] offer(1). docx ו[附件4] DiotData.xls 	
团全部打包加载	伍华聪的博客 http://wuhuacong.cnblogs.com
	确定取消

查看文件	×
BLUGERAY	图片预览对话框界面
	关闭
 ・「附件1] 3. jpg ×・「附件2] 4 test.jpg ×・「附件3] 0 offer(1).doox ×・「附件4] DictData.xls 	
□全部打包加载	
h	确定取消

2.12.1. 控制器后台的操作支持

删除文件的操作很简单,只需要根据附件的 ID (GUID 值),删除对应的文件,控制器 返回一个通用的操作结果对象即可完成。

```
/// <summary>
/// 删除单个附件信息
/// </summary>
/// <param name="id">附件的ID</param>
/// <returns></returns>
0 个引用
public ActionResult Delete(string id)
{
    CommonResult result = new CommonResult();
    if (!string.IsNullOrEmpty(id))
    {
        try
        {
            result.Success = BLLFactory<FileUpload>.Instance.Delete(id);
        }
        catch(Exception ex)
        {
            LogTextHelper.Error(ex);
            result.ErrorMessage = ex.Message;
        }
    }
    return ToJsonContent(result);
}
```

而文件的预览或者下载操作,是绑定到界面控件上的代码构建出一个 HTML 代码片段实现。

```
/// <summary>
/// 获取附件并生成相关展示的html,添加可删除附件链接。[Bootstrap样式的HTML]
/// </summary>
/// <param name="strGuid">附件组guid</param>
/// <returns>string</returns>
0 个引用
public ActionResult GetAttachmentBootstrap(string guid)
{
   string html = @"附件暂无";
   if (string.IsNullOrEmpty(guid))
       return Content(html);
   int seq = 1;
   StringBuilder sb = new StringBuilder();
   List<FileUploadInfo> fileList = BLLFactory<FileUpload>.Instance.GetByAttachGUID(guid);
   if (fileList != null && fileList.Count > 0)
   {
       构建附件展示的HTML代码
       return Content(result);
   }
   else
   {
       return Content(html);
   }
```

对于 Office 文档的在线预览,我们提供了两种方式,一种是在服务器本地利用组件生成 对应的 HTML 文件,然后预览;一种是通过微软的在线文档预览实现。第一种响应速度快, 不过文档有时候可能有些失真;第二种是需要服务发布到互联网,并且文档地址可以访问, 提供比较完美的展现,缺点是比较慢。

<pre>/// <summary> /// 根据附件ID,获取对应查看的视图URL。 /// 根据附件ID,获取对应查看的视图URL地址'/FileUpload/ViewAttach'; /// 如果是Office文件(word、PPT、Excel)等,可以通过微软的在线查看地址进行查看:'<u>http://view.officeapps.live.com/op/view.aspx?src=</u> /// 也可以进行本地生成HTML文件查看。如果是其他文件,可以直接下载地址。 /// </summary> /// // </pre>
/// <param name="10"/> pgr+pguo /// <pre>/// creturns></pre>
0 个引用 public ActionResult GetAttachViewUrl(string id) {
<pre>string viewUrl = ""; FileUploadInfo info = BLLFactory<fileupload>.Instance.FindByID(id); if (info != null) { string ext = info.FileExtend.Trim('.').ToLower(); string filePath = GetFilePath(info);</fileupload></pre>
bool officeInternetView = false;//是否使用互联网在线预览 string hostName = HttpUtility.UrlPathEncode(" <u>http://www.igidi.com/</u> ");//可以配置一下,如果有必要
if (ext == "xls" ext == "xlsx" ext == "doc" ext == "docx" ext == "ppt" ext == "pptx") { if (officeInternetView) Office在线预览
<pre>{ //返回一个微软在线浏览Office的地址,需要加上互联网域名或者公网IP地址 viewUrl = string.Format("<u>http://view.officeapps.live.com/op/view.aspx?src={0}{1}</u>", hostName, filePath); }</pre>
else {
} else {
<pre>viewUrl = filePath; } return Content(viewUrl); }</pre>

2.12.2. 界面层的代码

删除文件的 JS 代码如下所示。



其中 ShowUpFiles 就是用于刷新附件列表的处理,代码如下所示。



文件预览或者下载的 JS 处理代码如下所示。



其中使用/FileUpload/GetAttachViewUrl进行处理,获取真正的展示内容的URL,

主要是基于考虑,需要对 Office 文件进行 HTML 生成 (第一次生成,后续打开即可)。

如果是图片文件,这里会把图片的地址作为一个 URL,在独立窗口进行打开;如果是 其他附件,那么会自动下载到客户端。



下面是对 PPT 的预览效果。



2.12.3. 利用微软的平台进行 Office 文档的在线查看

在博客园很多文章里面,曾经有一些介绍Office 文档预览查看操作的,有些通过转为PDF 进行查看,有些通过把它转换为 Flash 进行查看,但是过程都是曲线救国,真正能够简洁方 第 91页 共 149页 便的实现 Office 文档的预览的还是比较少,这里的 Office 文档包括了 Word、Excel、PPT 文档。本文介绍两种方式,一种方式是通过在线预览的方式,利用微软的平台进行 Office 文档的在线查看;一种是把 Office 文档生成 HTML 文件后进行查看。

一直以来,都希望找一个合适的 Office 文档查看的控件或者解决方案,这样客户在使用 系统的时候,可以直接查看预览一些文档,而不需要安装 Office,或者下载到本地在查看。 一个偶然的机会,在网上搜到微软自己提供了一个在线服务,可以实现把 Office 文档转换为 在 线 的 内 容 进 行 查 看 , 大 家 可 以 通 过 下 面 的 链 接 地 址 大 致 看 看 : <u>http://blogs.office.com/2013/04/10/office-web-viewer-view-office-documents</u> <u>-in-a-browser/</u>。

使用浏览器直接查看 Office 文档的链接地址格式如下所示:

http://view.officeapps.live.com/op/view.aspx?src=http%3a%2f%2fvideo.ch9. ms%2fbuild%2f2011%2fslides%2fTOOL-532T_Sutter.pptx

这个链接分为了两部分,一部分是 http://view.officeapps.live.com/op/view.aspx?src=, 后 面那个是具体的文档地址,用 URLEncode 进行处理的链接地址。

非常酷的效果,使用这个服务唯一的要求就是你的网站是部署在互联网上的,这样服务才可 以调用你的文档地址然后进行展示。

这个是使用微软的预览 Office 服务,当然你可以部署自己的 Office 预览服务,也就是需要安装 Office Web Apps 服务(系统要求为 Windows Server 2012)

一般情况下,Office Web Apps 要与其他应用配合使用,如下图所示(看起来还是很复杂的,这些东西好像也要独立安装在不同的机器):



好在微软给我们提供了在线的 Office 文档预览服务,其实好像 Google Doc Viewer 也是可以的,但是已经不可使用了。这样,我们就可以利用公开的接口地址实现 Office 文档的在 线预览了。以常用的 Excel 文档为例,它可以提供了完美的在线预览效果。

x	XII Excel Online							
(Ì	牛 🕅 下載 数据 🕶 查找 批注							
	A	В	С					
1	Wedding Budget Planner							
2	Enter your estimated wedding budget and expenses to bett should make to plan your wedding within a budget that wor	er understand wi ks for you.	hat changes you					
3	If desired, insert new rows to include additional sources of do not enter any information in the blue rows. These cells h	funds or wedding old the formulas	expenses, but behind the chart.					
4	PART 1: Total Wedding Budget	Estimated	Actual					
5	Couple contribution							
6	Bride's family contribution							
7	Groom's family contribution							
8	Gifts							
9	Other							
10	TOTALS (Automatically Calculated)	\$0.00	\$0.00					

我们还注意到上面有一排菜单,可以展开进行相关功能的操作,尤其是文档的下载和打 印很吸引。



这样我们就可以在我们的文档预览操作页面上进行集成了,下面是 Web 开发框架集成的一个例子,对 Office 文档进行查看。

ſ	查	看附件		
1	X	I Exc	el Online art	?
	(文	件	【割下载 数据 ▼ 查找 批注	
н	-	A	В	
	1		客户导单;	A
	2		1.客户使用我们提供的固定模版导入数据到数据库。	
	3		2.客户执行合并按钮,把数据进行合并。	
	4		3.合并的同时,随机获取发件公司地址(从WG_FaAddress中随机获取)	
	5		4. 同时从WG_CompanyNoCity随机获取一个公司名, 前加 " 收件城市 " 填入 " 收件公司 "	
	6		6. 合开后的运单打印 这一个关系的法教员也	
	/		0. 百开后的这里守击 7. 如果方公法单导,刚体用,如果没方,刚相提报学公司,随机生成一么	
	9		了。如果有力运半亏,则使用,如果没有,则很掂扳大公可,随机主成 ^一 了。 	
	10	-	列伐扫描	
	11	_`	[1]选择某个λ合号后,扫描改λ合号的分法单号,有的해way ok way 如果没有刚way e	error1 1
	12		2.实时显示本批次一共多少个条码,扫描了多少,未扫描多少	
	13		(写)Dao Oper=当前田白 Dao ScanTime=当前时间)	•
		•		•
	м	4 F	M 并反技术 表说明 盖米 进度 WG_Bill WG_BillDetail_Cust	
			伍华聪的博客http://wuhuacong.cnblogs.com	• 業取消

Word 文档也是可以顺利进行预览,但是就没有打印和下载的功能了,不过预览的效果 还是很不错的。

查看附件		
	Word文档的在线查看效果	
	请您于 2014 年 5 月 1 日前回复确认接受此录用通知书并于 2014 年 6 月 1 日前携带 已签字确认的录用通知书到公司报到。	
	第一天入职时,您需要携带以下文件:	
	身份证(原件及复印件)	
	毕业证和学位证(原件及复印件)	
	一寸照片 (2张): 近期免冠白底彩色照片	
	医疗蓝本或社保卡	
	逾期未确认接受此录用通知书或逾期未报到,以及报到时不能提供上述真实有效文件的, 本通知书自始不发生法律效力。	
	其他说明:	
	1. 工作时间: 公司的工作时间为早晨 9 : 00 至下午 12 : 00 。中午 13 : 30 至 : 为	•
	伍华聪的博客http://wuhuacong.cnblogs.com	🎇 取消

2.12.4. 把 Office 文档生成 HTML 文件后进行查看

上面说了,使用在线的 Office 预览服务来查看 Office 文档,如果我们的管理系统是在局域网内跑的应用,那么我们就是用不了了,那么我们如果需要使用这种在线预览 Office 文档的服务,应该如何操作呢?

虽然自行搭建 Office Web Apps 服务应用可以解决这个问题,但是一般来说,搭建这样的平台环境,太过繁琐和昂贵了,有没有更好的方式实现 Office 文档的查看呢? 有的,下面我来介绍一下,如何使用 Aspose 组件把 Office 文档生成 HTML,然后进行查看的做法。

对应不同的 Office 文档, Aspose 提供了不同的组件, 如 Aspose.Word、Aspose.Cells、 Aspose.Slides 等不同的组件用来处理 Word/Excel/PPT 等几种文档。

我们知道, Aspose 组件在处理 Office 文档方面非常的强大, 但是也是收费软件, 所以需要可以购买支持, 但是我们这里纯粹讨论它的功能效果。

对不同的文件类型,我们调用不同的组件进行 HTML 的转换就可以了,核心部分代码 如下所示。

```
#region 动态第一次生成文件
//检查本地Office文件是否存在,如不存在,先生成文件,然后返回路径供查看
string webPath = string.Format("/GenerateFiles/Office/{0}.htm", info.ID);
string generateFilePath = Server.MapPath(webPath);
if (!FileUtil.FileIsExist(generateFilePath))
ł
    string templateFile = BLLFactory<FileUpload>.Instance.GetFilePath(info);
    templateFile = Path.Combine(System.AppDomain.CurrentDomain.BaseDirectory, templateFile.Replace("\\", "/"));
    if (ext == "doc" || ext == "docx")
    {
       Aspose.Words.Document doc = new Aspose.Words.Document(templateFile);
       doc.Save(generateFilePath, Aspose.Words.SaveFormat.Html);
    }
    else if (ext == "xls" || ext == "xlsx")
    {
        Workbook workbook = new Workbook(templateFile);
       workbook.Save(generateFilePath, SaveFormat.Html);
    }
    else if (ext == "ppt" || ext == "pptx")
    {
        templateFile = templateFile.Replace("/", "\\");
        PresentationEx pres = new PresentationEx(templateFile);
       pres.Save(generateFilePath, Aspose.Slides.Export.SaveFormat.Html);
    }
}
#endregion
```

这样,我们第一次使用的时候,判断目录里面是否有对应的 HTML 文件了,如果没有,就使用上面的代码生成就可以了,查看的时候,就返回对应的路径给客户端进行查看文件就可以了。

下面是几个文档的效果截图,供参考。

查看附件			使用Ap	oose.Cell生	成文件的预	览效果	
	河南机电高	高等专科学校联合办学董事单位通讯录 单位名称 次南机电高等专科学校 河南机电高等专科学校 河南金龙电缆集团公司 武 河南中原轴件集团公司 中原内燃机配件总厂 莽阳桦炜齿轮有限公司 高 安阳桃床集团有限责任公司 」					
	序号	单位名称	法人代表	联络员	任 1职 部 门	联系电话	单位
	1	河南机电高等专科学校	孙怀玉	卢杰	董事会 副秘书长	0373-3327938	新乡市北
	2	河南金龙电缆集团公司	武川	崔爱玲	厂工会主 席	0391-3933979 -6665	焦作市3
	3	河南中原轴件集团公司	马海洋	张振平	公司办	0391-3933051	焦作市麵
	4	中原内燃机配件总厂	薛德龙	姚守通	人劳处	0391-8190221	孟州市尊
	5	安阳桦炜齿轮有限公司	高泊依	陈宪宇	企业处主 任	0372-2923478-2050	安阳市ノ
	6	安阳机床集团有限责任公司	刘庆新	秦永昌	厂工会 副主席	0372-5924593-2223	安阳市角
4			14-34-7m		Designed to		→nn→±*
Sheet1 S	heet2 She	et3					
							🎇 取消

查看附件	
1.1.背景 使用Aspos	e.Word的预览效果
□ 《Winform开发框架》、《WCF开发框架》和《混合式开发框架	₽≫三个框架是我们经过多年的项目积累,吸收众多框架产品
客户的宝贵意见,反复提炼优化而成的。	
≪Winform开发框架≫用于传统的数据库通讯获取数据,这种	方式获取数据,方便快捷,可以用于常规的业务系统的场
景,用于单机版软件或者基于局域网内的业务系统软件。	
《WCF开发框架》采用了WCF技术的分布式开发模式,系统能够	9通过远程的WCF服务获取数据,而不用直接和数据库相连,
提高数据的安全性和印建护性,适用于 互联网、局域网 环境下的生 /泪合式工学标加入具地泪合了 <i>体统/Winger</i> 工学标加入和	业务系统的拾建,定一种稳定、安全的性保应用。 /wep开始框架》的特点,可以在西老文词白山切换,统工了。
◎ 凇山式开发框末/定组凇山」 传统 ◎ 細胞 0110开发框末/和 系统界面目动业各服务的调用模式。所有组件模块均实现两种方式	▼₩577友框来/的特点,可以在网有之间自由切换,统一」 代的调用,是一种弹性化非常好的框架应用,既可用于单机版
软件或者基于局域网内的应用软件,也可以用于分布式技术的互联	X0790/09/20 149年1270年8月1日10年8月2709 00/19701 平7000 美网环境应用。
1.2. 编写目的	
本文档主要介绍《Winform开发框架》、《WCF开发框架》和《	《混合式开发框架》中系统的功能特性,这三种框架除了它们
获取数据的万式不同外,他们具体大多数相同的功能特性,本文档	当王要看重介绍这三个基于插件化应用框架的各种相同特性。
12 会 字 次 約	
1.3. 参方页科	
席号 名称	版本/日期 来源
1 《Winform开发框架·架构设计说明书.doc》	内部
,	nblogs.com



2.13. Excel 数据导入导出

在我们常见的一些业务模块里面,往往 Excel 数据导入,是一个非常常见的功能,因此 在 Web 框架里面提供对 Excel 数据的导出和导出操作,也是很必要的一个功能点。

我们知道,Web上对 Excel 的处理和 Winform 的有所差异,如果是在 Web 上处理, 我们需要把 Excel 文档上传到服务器上,然后读取文件进行显示,所以第一步是实现文件

第 97页 共 149页

的上传操作,关于文件上传控件,具体可以参考上面小节的内容介绍。

Excel 数据的导入导出功能,在 Web 上的主界面如下所示。

▼査	询内容	Q 查询	试 导入	又 告 出	x	RDLC报表						
用户编码								真实姓名				
移动	动电话			邮件地址	ut 🗌		QQ号码					
◎\$数	✿ 数据列表 + 新增 ✔ 修改 田 查看 - 删除 C 刷新 X											
每页显	記 10	▼ 条记录	共有记录 : 461	条,总页数:	47 <mark>页。</mark>							
	D	用户编码	用户名/登录名	真实姓名	是否过期	是否删除	移动电话	邮件地址	性别		操作	
	462	shadmin	shadmin	上海管理员	正常	正常			男		۹ 🖉	×
	463	bjadmin	bjadmin	北京管理员	正常	正常			男		۹ 🖉	×
	464	gzadmin	gzadmin	广州管理员	正常	正常			男		۹ 🖉	×

导入界面如下所示。

文件导入					
				🛃 Use	er-模板.xls
浏览 上传 取消					
用户编码 用户名/登录名	真实姓名 职务头衔	移动电话 办公	电话 邮件地址	性别 00号	角 备注
				关闭	保存

文件	文件导入										
									1	Jser-模板. x1s	
 . [; [≹≦	減费 传 取 附件1] 王 て 全部打包加載	当 Jser (1)(1).xls			上示	传文件 在列表	后,Excel列君 中,勾选后可	€的 「以 ±	为容自 进行导,	动显入。	
	用户编 码	用户名/登录 名	真实姓名	职务头 衔	移动电话	办公电 话	邮件地址	性 别	99号码	备注	
	shadmin	shadmin	上海管理 员					男	6966265	上海分公司管 理员	
	bjadmin	bjadmin	北京管理 员					男		北京分公司管 理员	
	1001	admin	管理员		18620292076		wuhuacong@163.com	男	6966254	系统管理员	
	关闭保存										

导入操作,在 Bootstrap 框架里面,我把它作为一个层的,都统一放在 index.cshtml 文件里面,这样可以使得整个界面的处理更加紧密一点

另外,在导入导出的页面,以及后台控制器方法里面,绝大多数的代码逻辑已经准确生成,一般只需要进行一定的微调即可,这样可以减少很多编码的时间。

这里的文件上传处理,主要使用了 Uploadify 的这个控件进行处理的,当然也可以利用 我前面介绍过的 File Input 上传控件进行处理,都可以很好实现这些导入操作。一般情况 下的 Uploadify 控件的初始化代码如下所示。



2.13.1. 控制器后台的操作支持

为了有效处理数据的导入,我们需要严格保证导入的数据是和模板的字段是匹配的,否则处理容易出错,也没有任何意义。为了实现这个目的,框架里面提供方法对字段进行检查, 主要是确保 Excel 里面包含了完整的字段即可。



而文件检查通过后,我们会在服务器解析对应的 Excel 文件,把它 Excel 里面的数据转

换为 DataTable,最后初始化为实体类列表,并返回给调用页面就可以了。



最后就是把 DataGrid 里面的数据保存到服务器对应的业务表里面了,在控制器里面保存 业务对象的集合操作代码如下所示。

```
/// <summary>
/// 保存客户端上传的相关数据列表
/// </summary>
/// <param name="list">数据列表</param>
/// <returns></returns>
0 个引用
public ActionResult SaveExcelData(List<ContactInfo> list)
{
   CommonResult result = new CommonResult();
   if (list != null && list.Count > 0)
                                                           声明事务
   {
       #region 采用事务进行数据提交
       DbTransaction trans = BLLFactory<Contact>.Instance.CreateTransaction();
       if (trans != null)
       {
           try
           {
               //int seq = 1;
               foreach (ContactInfo detail in list)
                                                      遍历列表,插入数据库
               ł
                   //detail.Seq = seq++;//增加1
                   detail.CreateTime = DateTime.Now;
                   detail.Creator = CurrentUser.ID.ToString();
                   detail.Editor = CurrentUser.ID.ToString();
                   detail.EditTime = DateTime.Now;
                   BLLFactory<Contact>.Instance.Insert(detail, trans);
               }
               trans.Commit();
               result.Success = true;
           }
           catch (Exception ex)
           {
               LogTextHelper.Error(ex);
               result.ErrorMessage = ex.Message;
               trans.Rollback();
           }
       }
       #endregion
   }
   else
   {
       result.ErrorMessage = "导入信息不能为空";
   }
   return ToJsonContent(result);
}
```

而导出 Excel 数据的控制器方法,也主要就是生成一个 Excel 文件,供客户端下载即可。 具体的代码逻辑如下所示。



以上的代码,绝大部分都是通过代码生成工具,根据数据库的对应信息,进行动态生成 的,一般情况下,只需要进行适当的微调即可,非常方便。

2.13.2. 界面层的代码

为了实现 Web 上的数据导入导出操作,我们需要增加两个按钮,一个是导入按钮,一个是导出按钮。

BEGIN 数据查询
<pre><div class="portlet box green-meadow"></div></pre>
<pre><div class="portlet-title"></div></pre>
<pre><div class="caption"></div></pre>
<i class="fa fa-filter"></i> 查询内容
<pre><div class="tools"></div></pre>
 <button class="btn btn-circle btn-sm green" onclick="Refresh()" type="button"> <i class="fa fa-search"></i> 本 治</button>
<pre></pre> <pre><</pre>
zi class="fa_fa_fa_fa_fa_rit_o"x/ix
导入
<pre><button class="btn btn-circle btn-sm green" onclick="ShowExport()" type="button"></button></pre>
<pre><button class="collapse" javascript:;"="" onclick="javascript:locat</td></tr><tr><td></br></td></tr><tr><td></button></pre>

上面代码里面,对应的 JS 代码如下所示。



导出的 Excel 界面效果如下所示。

功能菜单 📃	系统用户信息	
公 首页	▼用户分类	▼查询内容
() 行业动态	按组织机构查看 按角色查看	用户编码
会员管理	□ 品) 州分公司 □ 營 总经办 □ 營 财务部	移动电话
客户管理		✿数据列表
← 权限管理	上 ▲ 测试组 □ 營 市场部 导出的用户表数据Excel文件	每页显示 10 ▼ 条记3
User (2).xls		,

第 104页共 149页

XI .	⊟ 5 ∗ ♂ ÷ ≑				(Contact (7).xl	s [兼容模式] -	Exce	1			?	个		- ×
文件	开始	插入	页面布局	公式	数据	审阅	视图 负载	测试	团队					登	₹ <u>0</u>
]
G6	*	$: \times$	$\sqrt{f_x}$												~
A		В		С	D	Е	F	G	н	1	J	K		L	-
1 序号	1	客户名	称	编号	姓名	身份证号码	出生日期	性别	办公电话	家庭电话	传真	联系人手机		联系人	地
2 1	广州爱奇训	自软件科技	支有限公司	1003	李经理		1900-1-1	男				1382515338			
3 2	广州爱奇训	自软件科技	支有限公司	1002	邱小姐		1986-11-14	女	87207161			1862029202			
4 3	广州爱奇训	自软件科技	支有限公司	1001	伍华聪		1978-10-1	男	87207160			1862029207	广州市	白云区尾	和
5															
6															
7									[
-	•	Sheet1	+						4						•
就绪									Ħ	B				-+ 10	0%

导入数据的时候,原理就是先展示需要导入的数据在界面上,用户勾选具体的记录后,

把对应的信息,添加到集合里面,统一执行方法进行写入就可以了。



写入操作如下所示。



为了更进一步获取用户导入到具体的部门,那么我们还可以弹出一个对话框用然后选择 具体的信息,最后才提交数据到后台进行处理。

		100750	100750	王璐琳	18601017599		Ę			
	选择	5年入用户	的公司和	部门						×
	所属公司* 默认部门*) 广州分公司		~			
)总经办			~		
									協会	TINE
ļ		400762	400762		40604240020		+	-	RHIZE	4000
		100702	100702	口天归	10001210029		>			
		100763	100763	杨梅	13681463018		身	3		
		100764	100764	刘洋波	13811991331		₽	3		

2.14. 图标样式生成和选择

在很多 Web 系统中,一般都可能提供一些图标的选择,方便配置按钮,菜单等界面元 素的图标,从而是 Web 系统界面看起来更加美观和协调。但是在系统中一般内置的图标样 式相对比较有限,而且硬编码写到样式表里面,这样给我们扩展使用有很多的不方便。基于 这个原因,我想如果能够独立一个模块,自动根据图标生成图标 CSS 样式文件,并存储相 应的记录到数据库里面,方便我们查询显示,那样我们使用起来就很方便了,最后有了这些
数据,只需要做一个通用的图标选择界面,并可以在很多地方重用了。

Web 框架提供了一个通用的图标生成、图标选择管理模块,非常适合于动态设置菜单、 按钮的图标操作,系统提供了一个对菜单图标样式动态配置的案例。

2.14.1. 菜单的显示及各种 Bootstrap 图标

我们从下图可以看到,为了菜单的美观,每个菜单项(这里分了三级菜单)都有一个图标,虽然大小不同,我们利用 Bootstrap 的图标,都是从 Bootstrap 图标库里面的内容。



Bootstrap 图标库里面分为了三类内容:

Font Awesome: Bootstrap 专用图标字体, Font Awesome 中包含的所有图标都 是矢量的,也就可以任意缩放,避免了一个图标做多种尺寸的麻烦。CSS 对字体可以设置 的样式也同样能够运用到这些图标上了。

如下面是部分 Font Awesome 的图标:

							1
迁	ム いんし 応田的り	<u>ह</u> ा-	上 二				
但	口 WED 应用的的	<u>ع</u> ار					
0	icon-adjust	Ø	icon-edit	Y	icon-magic	C	icon-share
*	icon-asterisk		icon-envelope	U	icon-magnet	~	icon-share-alt
Ø	icon-ban-circle	\sim	icon-envelope-alt	9	icon-map-marker	E	icon-shopping-cart
,II	icon-bar-chart	₽	icon-exchange	-	icon-minus	.al	icon-signal
	icon-barcode	0	icon-exclamation-sign	•	icon-minus-sign		icon-signin
≖	icon-beaker	ľ	icon-external-link		icon-mobile-phone	•	icon-signout
4	icon-beer	Þ	icon-eye-close	0	icon-money		icon-sitemap
4	icon-bell	۲	icon-eye-open	+	icon-move	\$	icon-sort
	icon-bell-alt		icon-facetime-video	5	icon-music	•	icon-sort-down

Simple Icons: 收集众多网站的 Logo,并提供高质量、不同尺寸的 png 格式图片给 广大网友,所有 Icon 版权归其所属公司。

如下面所示是 Simple Icons 的部分图标:

⚠ icon-user	A icon-user-female	री icon-users
Ω icon-user-follow	Ω icon-user-following	$\Omega_{\mathbf{x}}$ icon-user-unfollow
${f Y}$ icon-trophy	${igodol}$ icon-speedometer	▶ icon-social-youtube
${\mathfrak V}$ icon-social-twitter	៥ icon-social-tumblr	ੳ icon-social-facebook
😵 icon-social-dropbox	🛞 icon-social-dribbble	🕄 icon-shield
icon-screen-tablet	icon-screen-smartphone	🖵 icon-screen-desktop
icon-plane	icon-notebook	🗢 icon-moustache

Glyphicons:包括 200 个符号字体格式图表集合,由 Glyphicons 提供,Glyphicons Halflings 一般是收费的,但是经过 Bootstrap 和 Glyphicons 作者之间的协商,允许开发人员不需要支付费用即可使用。

如下是部分 Glyphicons 内容:

Icon glyphs			
下面展示的140个图标均提供	了深灰色(默认)和白色两种颜色	色。由 <mark>Glyphicons</mark> 提供。	
▼ icon-glass	🎜 icon-music	Q icon-search	🐱 icon-envelope
icon-heart	★ icon-star	🖈 icon-star-empty	L icon-user
目 icon-film	icon-th-large	III icon-th	icon-th-list
🖌 icon-ok	X icon-remove	Q icon-zoom-in	Q icon-zoom-out
🖒 icon-off	,,, icon-signal	icon-cog	💼 icon-trash
♠ icon-home	icon-file	O icon-time	A icon-road
🛓 icon-download-alt	icon-download		icon-inbox
O icon-play-circle	C icon-repeat	C icon-refresh	icon-list-alt
🖴 icon-lock	I [■] icon-flag	∩ icon-headphones	Icon-volume-off
 icon-volume-down 	 icon-volume-up 	鬷 icon-qrcode	🛄 icon-barcode
🗣 icon-tag	እ icon-tags	icon-book	icon-bookmark

利用这几种图标内容,我们引入下面几种样式就可以了。

<link href="/Content/metronic/font-awesome/css/font-awesome.min.css" rel="stylesheet"/>
<link href="/Content/metronic/simple-line-icons/simple-line-icons.min.css" rel="stylesheet"/>
<link href="/Content/metronic/bootstrap/css/bootstrap.min.css" rel="stylesheet"/>

这几种图标,都是支持各种 Bootstrap 的主题化显示的,如下面几种效果所示。

Default 🌣	Red 🚞	A Blue	Green 籠			
Yellow 🖍	💊 Purple	Green	9	Dark 🔗		





或者也可以把图标变大一些:



2.14.2. 各种 Bootstrap 的图标的提取

我们通过上面的介绍,估计对这几种 Bootstrap 的图标有了一定的了解,但是我们如果要能够在菜单编辑里面选择图标,那么我们还是需要把这些信息提取到数据库里面,然后展示出来给我进行选择的,否则无法做到动态配置。

文件名称: Web 框架界面代码分析

⊘ 修改信息		×
父菜单*	🖹 客户管理	×
系统编号*	📄 企业信息化平台	~
显示名称 *	📄 客户信息管理	
菜单是否可见	不可见	
排序	000	
功能ID	功能ID	提供图标动态选择
Web链接地址	/Customer/index	
Web图标	选择图标	
		确定取消

如上面的编辑界面,对菜单的 Web 图标提供了动态的选择,那么我们如果数据库里面 记录了上面几种图标的集合,那么我们就可以把它在界面进行展示,并可以从中选择合适的 图表了。

根据上面的几个样式文件,我们分析一下,如对于 font-awesome.min.css 的文件内容, 它对于图标定义部分如下所示。



而对于 simple-line-icons 来说,它的样式定义也差不多,如下所示。



对于 Glyphicons 来说,它的样式定义也是很类似的,如下所示。

文件名称: Web 框架界面代码分析



根据这几种信息,我们就可以通过正则表达式匹配的方式,把我们所需要的信息提取出

来,并存储在数据库里面即可实现图标动态显示和选择的第一步了。

例如,我们定义部分变量和正则表达式来处理这些文件内容:



最后我们把这些信息保存到数据库表里面即可。



最后记录存储在数据库里面,效果如下所示,里面已经记录我们所需的图标信息了,这

样在实际使用的时候,就可以利用各个字段的信息,显示出好看的界面了。

	ID	DisplayName	ClassName	SourceType	CreateTime
1	000cc33b-82cb-41a7-b9	glyphicon-plus	glyphicon glyphicon-plus	Glyphicons	<u> </u>
2	0101ab36-ac9b-45e3-bd	fa-caret-down	fa fa-caret-down	FontAwesome	2010-07-10 20.40.40.033
3	0122ace9-1934-438f-b8b	glyphicon-step-backward	glyphicon glyphicon-step-backward	Glyphicons	2015-07-15 20:43:44.673
4	016ccb0c-60d0-46ba-93	fa-users	fa fa-users	FontAwesome	2015-07-15 20:43:49.253
5	01912677-62d8-4652-b5	fa-sort-amount-desc	fa fa-sort-amount-desc	FontAwesome	2015-07-15 20:43:51.697
6	01d62ca3-467f-4a70-bdc	icon-shield	icon-shield	SimpleLine	2015-07-15 20:43:54.557
7	02068553-868d-4723-8f5	fa-wheelchair	fa fa-wheelchair	FontAwesome	2015-07-15 20:43:52.443

2.14.3. Bootstrap 的图标显示和选择

我们通过文件读取并以正则表达式提取出内容,然后保存到数据库后,这些图标信息就 可以为我们使用了,可以在页面里面分类显示出来,每类的图标进行分页处理,方便查询, 如下所示。

文件名称: Web 框架界面代码分析

▼ 图标分类	▼ 图表选择	示例										
└──È FontAwesome └──È Glyphicons └──È SimpleLine	Web图标 [] icon-screen-tablet 选择图标											
	▼ 图标信息	l.										
	每页显示 100 ▼ 条记录 共有记录: 0条,总页数: 0页。											
	F	\Leftrightarrow) 	ß	Ē.	Ē		(1)		Ø	<u>A</u>	Q
	Ę	Ç	1	R		\bigcirc	Ħ	\sim	Q	Ø		Q
	ŵ	ţ	Ċ	\bowtie	Ø	\odot	000	≺×	Ð	ŝ	(i)	
		57	G	Γ	\$	\$	Ĵ	\triangleleft	\bigotimes	€	0	ଦ୍ୱ
			£	69	ቲ			Ľ	P	0	ß	\diamond
	~??	f	ス ユ に ビ	Ē	R	O [™]	j	ß	A	444	Х,	(f)
	\sim	$\langle \rangle$	Q	\bigotimes	字	¢	\triangleright	ŧ	\heartsuit	7° 5		Ë
	\otimes	Q,	Ô	្រា	(+)	+- ×=	<u>D</u> *	\bigtriangledown	•			
	Ŕ	Û	(a)	\bigcirc								
	1 2	> >>										

这部分的显示页面代码和常规的数据显示差不多的,只是不需要表头信息而已,我们来看看 页面代码如下所示。

<div class="portlet-body flip-scroll"></div>
<div></div>
每页显示
<pre><select id="rows" onchange="ChangeRows()"></select></pre>
<option>50</option>
<pre><option selected="">100</option></pre>
<option>200</option>
<option>1000</option>
条记录 enbsp;enbsp;
共有记录: O 条,总页数:
<hr/>
<div class="row" style="padding-left:20px"></div>
<div class="portlet-body" id="grid_body"></div>
<pre><div class="paging-toolbar"></div></pre>
<ul id="grid_paging">

动态获取并生成 HTML 代码显示在界面上的处理脚本如下所示。



用户选择对应的图标后,我们可以通过脚本设置 Span 的样式就可以显示出来我们选中的图标了,如下所示。

\$("#i_WebIcon").attr("class", classname);

当然我们选择图标的时候,提供一个弹出的对话框显示分类不同的图标,让用户选择后 返回即可。

■ 查看图标信息 ×										
SimpleLine FontAwesome Glyphicons										
₹¥	¢	3 K	ß	f	Ē		(1)		Ø	
Æ	\mathcal{O}	Ļ	Ç	53	$\mathscr{C}_{\mathbb{K}}$		\bigcirc	甾	\sim	
Q	\triangleleft		Q	ᡬ	ÿ	Ċ	\bowtie	Ø	\odot	
000	⊲×	Þ	ŝ	i			5ª	G	C7	
\oplus	\$	Ĵ	\checkmark	\bigotimes	❹	0				
1 2 3	3 4 >	>>								
									_ ۲	ίŦ
										ا ه

这样我们就完成了,从图标文件里面提取不同类型的图表,然后存储在数据库里面,并 在页面里面显示出来,可供我们动态选择和设置了。

2.14.4. 控制器后台的操作支持

通过正则表达式对文件进行提取,并保存到列表里面,供保存数据库使用。

文件名称: Web 框架界面代码分析

```
/// <summary>
/// 生成图标文件
/// </summary>
/// <returns></returns>
0 个引用
public ActionResult GenerateIconCSS()
ſ
    CommonResult result = new CommonResult();
    string regex = "^\\.(?<name>.*?):before\\s*\\{";
    List<string> filePathList = new List<string>()
    {
        "~/Content/themes/metronic/assets/global/plugins/bootstrap/css/bootstrap.css",
        "~/Content/themes/metronic/assets/global/plugins/font-awesome/css/font-awesome.css",
        "~/Content/themes/metronic/assets/global/plugins/simple-line-icons/simple-line-icons.css",
    };
//图标类型
    List<string> sourceTypeList = new List<string>()
    {
        "Glyphicons",
        "FontAwesome"
        "SimpleLine",
    };
    List<string> prefixList = new List<string>()
    {
        "glyphicon ",
        "fa ",
        <u></u>,
    };
```

提出好数据后,就是逐一写入数据库对象里面了。

```
string prefix = prefixList[i];
string content = FileUtil.FileToString(realPath);
List<string> matchList = CRegex.GetList(content, regex, 1);
foreach (string item in matchList)
{
   //包含特殊的字符,去掉即可
    if (item.IndexOfAny(new char[] { '.', '>', '+' }) > 0)
        continue;
    BootstrapIconInfo info = new BootstrapIconInfo()
    {
        DisplayName = item,
        ClassName = prefix + item,
        CreateTime = DateTime.Now,
        SourceType = sourceType,
    };
    BLLFactory<BootstrapIcon>.Instance.Insert(info);
```

上面是对图标生成的相关操作,为了展示现有的图标信息,我们还需要根据条件从分页 数据库里面获取对应的图标对象信息,方便我们展示在界面上。

```
21 个引用
public override ActionResult FindWithPager()
ł
   //检查用户是否有权限,否则抛出MyDenyAccessException异常
   base.CheckAuthorized(AuthorizeKey.ListKey);
   string where = GetPagerCondition();
   PagerInfo pagerInfo = GetPagerInfo();
   List<BootstrapIconInfo> list = baseBLL.FindWithPager(where, pagerInfo);
   //如果需要修改字段显示,则参考下面代码处理
   //foreach(BootstrapIconInfo info in list)
   //{
   11
         info.PID = BLLFactory<BootstrapIcon>.Instance.GetFieldValue(info.PID, "Name");
         if (!string.IsNullOrEmpty(info.Creator))
   11
   11
        {
   11
             info.Creator = BLLFactory<User>.Instance.GetFullNameByID(info.Creator.ToInt32());
         }
   11
   //}
   //Json格式的要求{total:22,rows:{}}
   //构造成Json的格式传递
   var result = new { total = pagerInfo.RecordCount, rows = list };
   return ToJsonContentDate(result);
}
```

2.15. 统计图表 Highcharts 使用

在我们做各种应用的时候,我们可能都会使用到图表统计,以前接触过一些不同的图表 控件,但图表控件 Highcharts,其强大的功能和丰富的互动效果,令人难以忘怀。本篇主 要介绍在 Web 开发中使用图表控件 Highcharts,以及对其进行统一汉化等操作,让我们 的程序功能更加丰富,内容更加美观。

统计图表是很多应用程序需要拥有的功能,为了更好展示图表的使用操作,框架提供了 多种样式的图表演示。



2.15.1. Highcharts 基础介绍

Highcharts 是一个非常流行,界面美观的纯 Javascript 图表库。它主要包括两个部分: Highcharts 和 Highstock。Highcharts 可以为您的网站或 Web 应用程序提供直观,互动式的图 表。目前支持线,样条,面积, areaspline,柱形图,条形图,饼图和散点图类型。Highstock 可以为您方便地建立股票或一般的时间轴图表。它包括先进的导航选项,预设的日期范围, 日期选择器,滚动和平移等等。

HighChartS 官网: <u>http://www.highcharts.com/</u>

HighCharts Demo: http://www.highcharts.com/demo/

Highcharts 支持曲线图、饼图、柱状图、仪表图、散点图等等几十种图形,界面展示效果非常丰富,3D效果也很好看。列出几个供参考下吧









Highcharts 使用 jQuery 等 Javascript 框架来处理基本的 Javascript 任务。因此,在使用 Highcharts 之前,需要在页面头部引用这些脚本文件。如果你使用 jQuery 作为基本框架,那 么你需要在页面头部同时引用 jQuery 和 Hightcharts 两个文件就可以了。

第 120页共 149页

2.15.2. 控制器后台的操作支持

大多数情况下,我们都是通过动态构造数据的方式,提供数据给图表控件,实现图表的展示的,这种一般通过 JSON 数据进行传递,这里以集团公司人员和部门组成为例,构建用于图表展示的 JSON 数据。

集团人员分布情况数据生成如下:



部门分布情况的数据生成如下。

文件名称: Web 框架界面代码分析



2.15.3. 界面层的代码

使用图表控件 Highcharts 的时候,需要确认 JS 引用正确,如下所示。

```
@*脚本引用放在此处可以实现自定义函数自动提示*@
<script src="~/Scripts/CommonUtil.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-more.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script>
<script src="~/Content/JQueryTools/Highcharts/js/highcharts-d.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></sc
```

我们来分析下一个饼图的 Demo 代码,其中图表的 Series 数据,主要就是字典集合的

JSON 格式,因此我们可以通过动态从后台获取对应的数据,从而实现我们动态的图表了。

```
series: [{
   type: 'pie',
   name: 'Browser share',
   data: [
        ['Firefox',
                     45.0],
        ['IE',
                     26.8],
        {
            name: 'Chrome',
            y: 12.8,
            sliced: true,
            selected: true
       },
        ['Safari',
                     8.5],
        ['Opera',
                     6.2],
        ['Others',
                   0.7]
    ]
}]
```

如我一般倾向于使用 Jquery 的 Ajax 方式,调用后台获得数据,然后进行绑定的。首先我们 先在脚本函数里面,初始化一个 chart 对象,并把其中涉 series 数据 data 设置为空就是 了。

```
//初始化对象
$(function () {
   var chart1 = new Highcharts.Chart({
       chart: {
           renderTo: "container1",
           plotBackgroundColor: null,
                                              初始化一个饼图对象,数
           plotBorderWidth: null,
                                              据后面通过动态方式赋值
           plotShadow: false,
       },
       title: {
           text: '集团分子公司人员组成'
       },
       tooltip: {
           pointFormat: '{series.name}: <b>{point.y}</b>'
       },
       plotOptions: {
           pie: {
              allowPointSelect: true,
               cursor: 'pointer',
               dataLabels: {
                  enabled: true,
                  format: '<b>{point.name}</b>: {point.percentage:.1f} %',
                  style: {
                      color: (Highcharts.theme && Highcharts.theme.contrastTextColor) || 'black'
                  }
               },
               //showInLegend: true
           }
       ÷.,
       series: [{
                                          先以空白集合初始化
           type: 'pie',
name: '人员数量',
           data: []
       }]
```

第二步是通过 Ajax 调用后台连接获得数据,然后绑定到具体的属性上就可以了,具体

代码如下所示。

```
//通过Ajax获取图表1数据
$.ajaxSettings.async = false;
var data1 = [];
$.getJSON("/User/GetCompanyUserCountJson", function (dict) {
    for (var key in dict) {
        if (dict.hasOwnProperty(key)) {
            data1.push([key, dict[key]]);
        }
    };
    chart1.series[0].setData(data1);
});
```

而图表的 HTML 代码则是如下所示,主要就是新增一个 div, id 为 container1,用来

放置图表就是了。

2.16. 标签控件 Tags-Input

我在利用 jQuery Tags Input 插件之前,一直想找一个合适的 Jquery 插件或者合适的做法,来实现我 Winform 框架里面权限系统的一个用户选择场景,就是能够记录用户的选择,并最终能够保存到数据库里面去。在 WInform 里面,我可以用自定义控件的方式,很好地实现了这个功能,但是在 Web 界面上,我尝试用 Jquery 试过了很多方法,没能实现这个效果,花了不少时间来寻找,终于找到这个不错的插件。

先来看看我的最终实现的 Web 界面效果,就是在权限管理系统里面,机构包含的用户 编辑界面,或者是角色包含人员的编辑界面里面,提供一个地方来记录用户的选择,用户确 认后,可以把记录的内容保存到数据库里面。

用户分类	« 🔲	系统	用户信息						
晶 按组织机构查看 🛛 🥵 按角色查看	-)添加	选择 🖌 🗸 完	成选择 🔲 清潔	2 💈 刷新				
1 🔐 爱奇迪集团			ID	用户编码	用户名/登录名	真实姓名	是否过期	职务头衔	移动电读
▲ 矗 广州分公司	1		462	shadmin	shadmin	上海管理员	√ त⊞		
◎● 尽空の ◎● 尽空の	2		463	biadmin	biadmin	北京管理员	v √ ਹ ⊞		
25 工程部	3		464	gzadmin	ozadmin	广州管理员			
	4		18	100001	100001	「たち」			138101
	5		445	100002	100003				125755
A 测试组	5		02	100003	100003	아니슈~티 기지击	♥ □用		120170
4 🍰 市场部	-		93	100004	100004	か开	♥ 可用		139170
A 市场一部			145	100006	100006	東院当	✔ 可用		186/40
San an a	8		350	100009	100009	土景景	✔ 可用		137616
2014年二日 1月11日 1月111日 1月1111111 1月11111111	9		53	100012	100012	朱丽华	🖌 可用		136612
◎ 人力资源部	10		438	100015	100015	王伟	🖌 可用		133707
2 ····· 上海方公司 然 财务部	11		127	100018	100018	杨维艳	🖌 可用		139963
🍰 市场部	12		427	100020	100020	包英浩	✔ 可用		186280
▲ 品 北京分公司	13		437	100021	100021	黄志辉	🖌 可用		186966
🎒 财务部 🥵 市场部	14		21	100022	100022	周黎品	✔ 可用		134523
	15		451	100023	100023	邱海维	✔ 可用		131023
		•		m					
	1	5 🖵	14 4	筆1 共31页	▶ н с	5		显示1到15.	.#465记录
选择的用户									3
		10 ma	74/20	2021 min					
官埋页 X 李陈佳 X 齐景琪 X	日萍 X	胡欣	X St静春 X	(梁爽 X 包	英活 X 土珣	X 岁雪 X			

上图下面一个区域"选择的用户"里面就是我用到这个控件来展示用户选择的人员信息。其实这个 jQuery Tags Input 插件主要的用途,是用来记录用户输入的标签的,它可以在空白的地方接受输入的内容的,如下所示。



2.16.1. Tags Input 插件的使用

由于在 MVC 项目里面中集成使用,因此你需要整理好合适的路径,我的项目代码引用的路径如下所示。

```
<script src="~/Content/JQueryTools/Tags-Input/jquery.tagsinput.js"></script>
<link rel="stylesheet" type="text/css"
href="~/Content/JQueryTools/Tags-Input/jquery.tagsinput.css" />
```

不过由于在框架里面已经统一使用了 BundleConfig 来实现对控件 JS 和 CSS 文件的引

入,实际上上面的代码不需要在视图 HTML 代码中出现。

简单的例子就是在需要的表单里创建一个包含 tags 列表的 input 输入框,你可以在 value 第 125页 共 149页

里设置默认或目前有的 tags,并用逗号隔开。

<input name="tags" id="tags" value="foo,bar,baz" />

这个插件可以屏蔽界面上的 Tag 标签输入,从而让脚本根据需要写入不同的标签。可以使用 addTag() and removeTag()函数增加和删除掉标签,代码如下:

```
$('#tags').addTag('foo');
```

```
$('#tags').removeTag('bar');
```

还可以用 imporTags()方法导进一组 tag 列表,需要注意的是这样会将 value 里设置 tag 替换掉。

```
$('#tags'). importTags('foo, bar, baz');
```

如果传递参数为空,那么相当于清空列表了。

```
$('#tags').importTags('');
```

使用 tagExist()可以判断一个标签是否存在:

```
if ($('#tags').tagExist('foo')) { ... }
```

这个插件还可以接受自动提示的插入操作,如下所示。

```
$('#tags').tagsInput({
    autocomplete_url:'http://myserver.com/api/autocomplete'
```

});

如果想要在增加或移除掉标签的时候增加额外的功能或触发其它动作,你可以通过 onAddTag和 onRemoveTag 这两个参数里指定回调函。这两个函数都返回了一个标签值作为 参数.

```
$('#tags').tagsInput({
    width:'auto',
    onAddTag:function(tag){
        console.log('增加了'+tag)
    },
    onRemoveTag:function(tag){
        console.log('删除了'+tag)
    });
```

前面讲了,可以屏蔽界面的 Tag 标签输入,而通过脚本插入标签,或者你想提供其它交 互方式增加标签,可以增加一个值为 false 的 interactive 参数,这样就禁止了增加标签,而 其它的功能和呈现都跟原来一样。

```
$('#tags').tagsInput({
    width:'auto',
    onRemoveTag:function(tag){
        console.log('移除标签: '+'"'+tag+'"')
```

```
},
interactive:false
});
```

2.16.2. 在项目中使用 Tags Input 插件

在项目需要使用的地方,添加一个 id=tags 的标签,如下所示。



我们先初始化列表和 Tags 标签对象,并增加一个添加用户的封装和移除用户的封装操

作,代码如下所示。

```
<script type="text/javascript">
      $(function () {
          $('#tags').tagsInput({
              width: 'auto',
              height: '100px',
              onRemoveTag: function (tag) {
                   var i = addNameList.indexOf(tag);
                   var id = addUserList[i];
                   removeUser(id, tag);
              },
              interactive: false
          });
      });
      var addUserList = new Array();
      var addNameList = new Array();
      function addUser(id, name) {
          if ($.inArray(id, addUserList) == -1) {
              addUserList.push(id);
              addNameList.push(name);
              $('#tags').addTag(name);
          }
      }
      function removeUser(id, name) {
          if ($.inArray(id, addUserList) != -1) {
               addUserList.pop(id);
              addNameList.pop(name);
              $('#tags').removeTag(name);
          }
      }
/script>
```

清除用户选择的 Tag 操作,代码也很简单了,前面介绍过了,熟练应用就是了。

```
//清空用户选择记录
function cleareChoise() {
   $('#tags').importTags('');
   addUserList = new Array();
   addNameList = new Array();
}
```

对于最重要的保存操作,就是把存储用户 ID 的列表,把他们传递给对应的 Ajax 调用 就搞定了。

```
//完成选择并返回
function completeChoise() {
   var ouid = $('#txtID').val();
   if (ouid != "") {
       var url = '/OU/EditOuUsers?r=' + Math.random();
       saveAction(url, ouid, addUserList);
   }
   $("#DivEditUser").dialog('close');
   reloadRelation();//重新刷新
}
//保存机构用户操作
function saveAction(url, id, newList) {
   $.ajax({
       type: 'POST',
       url: url,
       async: false,
       data: { ouid: id, newList: newList.join(',') },
       success: function (result) {
           $.messager.alert("提示","操作成功!");
           $('#DivEditUser').dialog('close');
           reloadRelation();
       },
       error: function (xhr, status, error) {
           $.messager.alert("提示", "操作失败"); //xhr.responseText
       }
   });
}
```

最后,我们就可以顺利看到真正的结果了。



2.17. RDLC 报表生成和预览

RDLC 是一个不错的报表,有着比较不错的设计模式和展现效果,在我的 Winform 开发里面,使用 RDLC 也是一个比较方便操作,如可以参考文章《<u>DevExpress 的 XtraReport</u> 和微软 RDLC 报表的使用和对比》或者《<u>会员管理系统的设计和开发(2)-- RDLC 报表的</u> 设计及动态加载》进行了解。但是基于 MVC 方式,一样可以构建和展现基于 RDLC 的报表。

🛃 企业信息化	P台管	理系统				当前时间	: 2014年11月20	日上午9:26:40	🔊 💽 🔀
管理员,您好!	行业动态	仓库管理	业务报表	客户管理	系统管理				
导航菜单 《	首页	业务报表	×						
■业务报表						伍华联	的博客http	//wuhuacong.	cnblogs.com 📤
品 业务报表 图 图本体注				系	统用户	记录报	表		
品 图表统计		累计人数	: 463		打印日期	2014-1	1-20 9:26		
🔜 月结操作		戊旱	田白槍和	田白夕/務忌夕	青空推交	基本计加	移动电话	新属公司	
晶 年度汇总报表		1	shadmin	shadmin	上海管理员	正常	1949/145 40	上海分公司	
晃 交续信白担子		2	bjadmin	bjadmin	北京管理员	正常		北京分公司	
🗰 术现信忌证小		3	gzadmin	gzadmin	广州管理员	正常		广州分公司	
		4	100001	100001	陈莉	正常	13810107123	广州分公司	
		5	100003	100003	阮琪琦	正常	13575509680	北京分公司	
		6	100004	100004	邓卉	正常	13917028898	广州分公司	
		7	100006	100006	黄晓雪	正常	18674042588	广州分公司	
		8	100009	100009	王景景	正常	13761601760	广州分公司	
		9	100012	100012	朱丽华	正常	13661246623	广州分公司	
		10	100015	100015	王伟	正常	13370717219	广州分公司	
		11	100018	100018	杨维艳	正常	13996351507	广州分公司	

2.17.1. 控制器后台的操作支持

绑定 RDLC 报表,并赋值对应的数据源操作如下所示。

```
0 个引用
public ActionResult RDLCReport()
{
    return View("RDLCReport");
}
/// <summarv>
/// 基于RDLC的报表数据操作
/// </summary>
/// <param name="format">图片格式</param>
/// <returns></returns>
0 个引用
public ActionResult UserRdlcReport(string format)
{
    LocalReport localReport = new LocalReport();
    localReport.ReportPath = Server.MapPath("~/Report/WHC.UserReport.rdlc");
    var dt = baseBLL.GetAll();
    ReportDataSource reportDataSource = new ReportDataSource("DataSet1", dt);
    localReport.DataSources.Add(reportDataSource);
    if(string.IsNullOrEmpty(format))
    {
        format = "Image";
    }
    string reportType = format;
    string deviceType = (format.ToLower() == "image") ? "jpeg" : format;
    string mimeType;
    string encoding;
    string fileNameExtension;
```

呈现的操作代码如下所示,默认我们以图片进行展现。

```
Warning[] warnings;
string[] streams;
byte[] renderedBytes;
renderedBytes = localReport.Render(
    reportType,
    deviceInfo,
    out mimeType,
    out encoding,
    out fileNameExtension,
    out fileNameExtension,
    out streams,
    out warnings);
return File(renderedBytes, (format.ToLower() == "image") ? "image/jpeg" : mimeType);
```

我们了解 RDLC 的话,应该知道,一般 RDLC 报表,它都是通过一个 DeviceInfo 的信息进行展现的,如下所示是一个标准的 DeviceInfo 对象。

```
string deviceInfo =
    "<DeviceInfo>" +
        " <OutputFormat>" + deviceType + "</OutputFormat>" +
        " <PageWidth>8.5in</PageWidth>" +
        " <PageHeight>11in</PageHeight>" +
        " <MarginTop>0.5in</MarginTop>" +
        " <MarginLeft>1in</MarginLeft>" +
        " <MarginRight>1in</MarginRight>" +
        " <MarginRight>" +
        " <MarginRight>" +
        "
```

" <MarginBottom>0.5in</MarginBottom>";

但是这样的内容,如果展现图片的话,就只会展示一页的内容,一般是 800 的高度这样 子,但是我的报表里面可能有很多记录,如何能够让它全部展现出来呢?

```
第 130 页 共 149 页
```

方法是有的,不过不是很完美,就是需要计算大概的尺寸,然后修改 PageHeight 的数 值,让它动态的删除最大的记录,达到全部内容都可以输出看到。

为了达到这个目的,我对图片格式输出的报表,对它的高度进行了一个简单的计算,然 后换成它的标准高度,这样代码如下所示。

```
if(format.ToLower() == "image")
{
    double inchValue = (dt.Count / 37.0) * 11;
    deviceInfo += string.Format(" <PageHeight>{0} in</PageHeight>", inchValue);
    else
    {
        deviceInfo += " <PageHeight>11in</PageHeight>";
    }
}
```

2.17.2. 界面层的代码



2.18. LODOP 打印模块的使用

2.18.1. LODOP 打印控件介绍

在 Web 框架里面,通知公告内容的打印预览界面如下所示,该模块集成了 LODOP 打印控件,因此预览效果非常美观,这个打印控件在很多场合表现出色,还可以用作证件的套打操作。

普通内容的打印界面如下所示:

5 打印预览	x
	^
关于做好防御今年第19号台风工作 的紧急通知	
发布者: admin 发布时间: 2013-09-26 16:13:24	
各区、县级市交通局、各行业协会:	=
今年第19号台风"天兔"21日14时中心位于深圳市偏东南方约730公里的巴士海峡海面上。据广东省气象 台预计, "天兔"将以20公里左右时達向西北偏西方向移动,将于22日下午到夜间以强台风级别(14级 左右)在惠来到台山之间沿海登陆,正面袭击珠江三角洲,登陆后继续西行横穿广东中西部,广州地区 将迎来狂风暴雨。	
诸各单位认真做好防御今年第19号台风工作,结合实际,采取措施,加强宜传和检查工作,督促停车场 企业加强地下停车库排水系统与设施的检查与调试,做好防、排水物资的准备,落实责任人,加强值 班,密切留意天气变化情况,做好雷雨大风等强对流灾害天气防御工作。	
特此通知。	
市交委	
2013年9/522日	
附件智无	
	-
*	r
5126.203.0mm~200.0mm	

证件套打的打印界面如下所示。



Lodop 是专业 WEB 控件,用它既可裁剪输出页面内容,又可用程序代码直接实现复杂

打印。控件功能强大,却简单易用,所有调用如同 JavaScript 扩展语句,主要接口函数如下:

- PRINT_INIT(strPrintTaskName) 打印初始化
- SET_PRINT_PAGESIZE(intOrient, intPageWidth, intPageHeight, strPageName) 设定纸张大小
- ADD_PRINT_HTM(intTop,intLeft,intWidth,intHeight,strHtml) 增加超文本项
- ADD_PRINT_TEXT(intTop,intLeft,intWidth,intHeight,strContent) 增加纯文本项
- ADD_PRINT_TABLE(intTop,intLeft,intWidth,intHeight,strHtml) 增加表格项
- ADD_PRINT_SHAPE(intShapeType,intTop,intLeft,intWidth,intHeight,intLineStyle, intLineWidth,intColor)画图形
- SET_PRINT_STYLE(strStyleName, varStyleValue) 设置对象风格
- PREVIEW 打印预览
- PRINT 直接打印
- PRINT_SETUP 打印维护
- PRINT_DESIGN 打印设计

2.18.2. 使用 LODOP 打印控件

在使用 LODOP 前,我们需要在页面中引入"LODOP/CheckActivX.js"文件,这个文件是对 LODOP 的检查并初始化对象的,这样我们在页面就可以直接使用全局对象进行打印等操作了。默认情况下,我们生成的视图界面里面,已经在包含了这个引用了(在 BundleConfig 里面)。

要实现 Web 界面内容的打印,主要是要构造待打印的 HTML 代码即可,一般情况下,

第 133页共 149页

我们可以把界面的某部分作为内容输出给打印控件,必要的时候,设置一下 HTML 代码的



样式,这样会使得界面和我们浏览器看到的结果一致。

一般的套打,包含了几部分操作:打印预览、打印维护、打印设计。

打印预览和打印维护是面向终端用户的,打印维护是指内容不能修改删除、但位置可以 调整,给不同的打印机不同的尺寸打印提供调整位置的可能性。

打印设计是面向开发人员的,开始需要通过这个功能来设计好套打的界面,就是根据套 打证件的背景图片,大致摆放好各个内容的位置。

在套打的时候,注意需要通过下面代码来设置显示背景图片(打印的时候,是不打印背 景的)。

LODOP. ADD_PRINT_SETUP_BKIMG("");

LODOP.SET_SHOW_MODE ("BKIMG_IN_PREVIEW",1); //打印预览时是否包含背景图 如下面的就是证件套打案例里面的实例代码。



上面的位置及文本内容的代码,是通过这个打印控件的设计模式进行调整和修改的,可 以在其中增加内容,也可以调整具体的文本位置,从而实现套打内容的设计,最后把固定好 的内容复制到页面上即可。



2.19. 富文本控件 CkEditor 和 Ckfinder 控件

Web 开发上有很多 HTML 的编辑控件,如 CKEditor、kindeditor 等等,很多都做的很好,本小节主要介绍在 MVC 界面里面,CKEditor 的配置和使用。

CKEditor 的前身是 FCKEditor,随着它的更新,上传图片的功能被分离出去了,现在如 果需要实现上传图片,要么自己写代码或者采用其他上传控件(如 Uploadify),还有一种方 法是使用 CKFinder,这两者的合并使用,能给我们带来更多的方便。

在利用代码生成工具生成的 Web 框架的视图代码里面,已经包含了对这个控件的应用, 以及提供了一个初始化的函数,可以在其中进行设置;或者参考行业动态、法律法规模块里 面对 CKEditor 的使用代码。

下面介绍的 CkEditor 和 Ckfinder 控件的使用知识,主要是基于对这个控件进行详细的 了解,以及重新处理新下载控件的设置参数等内容。

2.19.1. CKEditor 的使用

CKEditor 的下载地址是 http://ckeditor.com/download, 里面可以根据需要进行样

式的定制,Web框架主要介绍当前较新的版本 4.4.1 的继承使用。而 CKFinder 的下载地址是: http://ckfinder.com/download。

CKEditor 的使用比较简单,一般情况下根据官方的指引选择适当的样式下载就可以了,使用的时候,基本不需要怎么样修改配置文件。在 MVC 的视图页面里面,添加相关的引用文件就可以了。

<script src="~/Content/JQueryTools/ckeditor/ckeditor.js"></script>

<script src="~/Content/JQueryTools/ckeditor/adapters/jquery.js"></script>

然后在界面添加需要编辑 HTML 内容的 textarea 控件,由于我的使用的是 EasyUI 的控件,

因此设置了控件样式 class="easyui-validatebox",也可以不管这里。

```
<label for="Content">内容: </label><textarea class="easyui-validatebox" id="Content" name="Content"</td>style="width:1024px"></textarea>
```

我们一般使用的时候,需要一段 javascript 脚本进行初始化对应的控件,初始化代码如下所示。

```
<script>
  function initEditor() {
     $('#Content').ckeditor();//控件1
     $('#Content1').ckeditor();//控件2
  }
</script>
```

添加相应的脚本和控件初始化代码后,就可以在界面中呈现出需要的效果了。



而之后的控件使用,就和普通的使用没有两样了,如赋值语句如下所示。

\$('#Content1').val(info.Content);//ckeditor 赋值

获取控件的值,也和普通给的一样

```
var content = $("#Content1").val();
```

2.19.2. CKFinder 的集成使用

虽然 CKFinder 已经独立出来,但是它也提供了完美的整合效果,我们把它下载后,在 压缩包里面的 bin 目录里面找到相应的 ckFinder.dll,把它添加我们项目工程的引用里面去, 我们才能正常使用文件上传功能。

然后修改 config.ascx 文件里面的一些设置,使得我们能够顺利使用。

第一步设置 CheckAuthentication 函数返回 True。

```
public override bool CheckAuthentication()
{
    return true;
}
```

第二步是设置 SetConfig 函数里面的 BaseURL,这个基础地址需要设置和我们项目的地址 一致,否则上传文件有问题。

第三步,整合 CKFinder 到 CKEditor,前面说了 CKEditor 默认是没有文件上传的功能操作的,需要添加 CKFinder 并进行配置才可以使用。这步骤需要在 CKEditor 里面的 config.js

```
第 138页共 149页
```

文件里面修改下面的配置参数(红色部分就可以了)。

```
CKEDITOR.editorConfig = function( config ) {
```

 $//\ {\rm Define}$ changes to default configuration here.

- // For complete reference see:
- // http://docs.ckeditor.com/#!/api/CKEDITOR.config

config.filebrowserBrowseUrl = '/Content/JQueryTools/ckfinder/ckfinder.html'; //上传文 件时浏览服务文件夹

config.filebrowserImageBrowseUrl =

```
'/Content/JQueryTools/ckfinder/ckfinder.html?Type=Images'; //上传图片时浏览服务文件夹
config.filebrowserFlashBrowseUrl =
```

'/Content/JQueryTools/ckfinder/ckfinder.html?Type=Flash'; //上传 Flash 时浏览服务文件夹 config.filebrowserUploadUrl =

'/Content/JQueryTools/ckfinder/core/connector/aspx/connector.aspx?command=QuickUpload&ty pe=Files'; //上传文件按钮(标签)

config.filebrowserImageUploadUrl =

'/Content/JQueryTools/ckfinder/core/connector/aspx/connector.aspx?command=QuickUpload&ty pe=Images'; //上传图片按钮(标签)

```
config.filebrowserFlashUploadUrl =
```

'/Content/JQueryTools/ckfinder/connector/aspx/connector.aspx?command=QuickUpload&type=Flash'; //上传Flash 按钮(标签)

```
};
```

2.19.3. 集成效果展示

通过以上代码进行整合,在插入图片的操作页面里面,会增加一个浏览服务器按钮,上 传操作选项卡等功能,方便对文件的浏览和上传操作,具体效果如下所示。

图像属性	×
图像信息 链接 上传	
URL	
目标窗口 <没有设置> ▼	刻齿服务器 整合增加的按钮
伍华聪的博客 http://	wuhuacong.cnblogs.com
	衛定 取消

图像信息 链接 上传 增加的工作文件操作
上传到服务器 选择文件 web4026.png
上传到服务器
上传到服务器
伍华聪的博客 http://wuhuacong.cnblogs.com
The second se
<u> 确定</u> 取消 _



以上就是我在我的 Web 框架里面整合的 HTML 编辑控件和 CKFinder 文件上传组件, 这两个组合起来, 能够非常方便构建图文并茂的文章内容。

2.19.4. MVC 的处理

这里需要注意的是由于 textarea 中有特殊字符,出于安全原因,默认情况 mvc 框架不允 许提交的,应在相应方法上加上[ValidateInput(false)]属性。

如我为了提交 HTML 内容,需要在控制器对象里面,重写了内容的写入和更新操作函数,

如下所示。

```
[ValidateInput(false)]
public override ActionResult Insert(InformationInfo info)
{
    info.Editor = CurrentUser.Name;
    info.EditTime = DateTime.Now;
    return base.Insert(info);
}
[ValidateInput(false)]
public override ActionResult Update(string id, FormCollection formValues)
{
    return base.Update(id, formValues);
}
```

2.20. summernote 插件实现 HTML 文档的编辑和图片插入

在很多场合,我们需要在线编辑 HTML 内容,然后在页面上或者其他终端上(如小程序、APP 应用等)显示,编辑 HTML 内容的插件有很多,本篇介绍基于 Bootstrap 的 summernote 插件实现 HTML 文档的编辑和图片插入操作,这个控件的使用非常方便,并且用户群也很大。

Summernote 是一个简单灵活,所见即所得(WYSIWYG)的编辑器,Summernote 是一个轻量级、灵活基于 Bootstrap 和 jQuery 的 HTML 文本编辑器,拥有强大的 API 配 置功能,多国语言支持支持 Bootstrap2.x 和 3.0,支持视频和图片上传以及代码的高亮显 示,多种后台语言版本示例以及多主体支持,在浏览器兼容方面,支持 IE9+以及现代的浏 览器 Chrome,Firefox,Safari 等,在移动端应该有不错的表现,还是很不错的,值得使用。

该插件是开源的, 官网地址:<u>http://summernote.org/</u>, GitHub 地址: <u>https://github.com/summernote/summernote/</u>、安装调整都很方便。

2.20.1. Summernote 的简单使用

使用方法:

1)加载 JS 和 CSS

```
<!-- include libraries(jQuery, bootstrap) -->
<link href="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.css" rel="stylesheet">
<script src="http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
<script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
</script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
</script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
</script src="http://netdna.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.js"></script>
</script src="http://cdnjs.cloudflare.com/ajax/libs/summernote/0.8.7/summernote.css" rel="stylesheet">
</script src="http://cdnjs.cloudflare.com/ajax/libs/summernote/0.8.7/summernote.js"></script></script></script</script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script><
```

基于 MVC 的 Asp.net 应用程序中,我们一般在 BundleConfig.cs 里面初始化我们的引入

JS 文件,如下所示。

//添加对bootstrap-summernote的支持
css_metronic.Include("~/Content/metronic/assets/global/plugins/bootstrap-summernote/summernote.css");
js_metronic.Include("~/Content/metronic/assets/global/plugins/bootstrap-summernote/summernote.min.js");
js_metronic.Include("~/Content/metronic/assets/global/plugins/bootstrap-summernote/lang/summernote-zh-CN.min.js");
bundles.Add(css_metronic);
bundles.Add(js metronic);

2) 编写 HTML 内容
在 HTML 页面代码增加下面的标记。

<div id="summernote">Hello Summernote</div>

3) 初始化调用

简单的初始化代码如下所示。

```
$(document).ready(function() {
    $('#summernote').summernote();
});
```

```
如果增加控件高度的定义,则如下所示。
```

```
$('#summernote').summernote({
    height: 300,
    focus: true
});
```

而详细的初始化界面,一般还包括语言国际化、图片上传等处理代码,如下所示

```
function initEditor() {
   $('#Note').summernote({
       lang: 'zh-CN', // default: 'en-US'
       height: 600,
                         // set editor height
       minHeight: null, // set minimum height of editor
       maxHeight: null,
                        // set maximum height of editor
       focus: true,
                          // set focus to editable area after initializing summe
       callbacks: {
           onImageUpload: function (files) { //the onImageUpload API
              img = sendFile(files[0]);
           }
       }
   });
```

4) 获取内容

```
var markupStr = $('#summernote').summernote('code');
```

5) 设置内容

设置内容和获取内容类似,在 code 后面增加需要写入的 HTML 内容即可。

```
var markupStr = 'hello world';
$('#summernote').summernote('code', markupStr);
```

2.20.2. summernote 插件在实际项目中的使用

上面介绍了常规 summernote 插件的使用过程,一般情况下,我们使用它除了编辑 HTML 内容外,还需要对图片上传进行处理,以确保我们可以直接把文件上传到后台文件系 统里面去,而不是把它们转换为 Base64 的编码放在页面内。

一般的初始化代码如下所示。

<pre>function initEditor() {</pre>	
<pre>\$('#Note').summernote({</pre>	
lang: 'zh-CN',	// default: 'en-US'
height: 600,	// set editor height
minHeight: null,	// set minimum height of editor
maxHeight: null,	// set maximum height of editor
focus: true,	// set focus to editable area after initializing summe
callbacks: {	
onImageUpload:	<pre>function (files) { //the onImageUpload API</pre>
img = sendF	<pre>ile(files[0]);</pre>
}	
}	
});	
}	

其中的 sendFile 的函数如下所示,主要是调用附件管理模块进行文件的存储。

```
//提交文件到服务器处理
function sendFile(file) {
   data = new FormData();
   data.append("file", file);
   //増加额外的参数
   data.append("folder", '商品信息');
   data.append("guid", $("#ID").val());
   $.ajax({
       data: data,
       type: "POST",
       url: "/FileUpload/Upload",
       cache: false,
       contentType: false,
       processData: false,
       success: function (json) {
           var data = $.parseJSON(json);
           var url = data.urls[0];
           $("#Note").summernote('insertImage', url, 'image name'); // the insertImage API
       }
    });
```

一般来说,我们的文件为了在多个应用之间共享处理,一般建议以 FTP 方式进行处理,这 方面可以参考随笔《<u>在附件管理模块中增加对 FTP 上传和预览的支持</u>》,FTP 上传文件可 以使用 FluentFTP 这个组件(GitHub 地址:

<u>https://github.com/hgupta9/FluentFTP</u>),这个是一个应用很广,功能很强大的 FTP 组件。

上传附件到服务器上的后台控制器代码如下所示,主要是构建文件信息,上传后返回对应的 URL,然后就可以在 summernote 插件上显示图片了。

```
/// <summary>
/// 上传附件到服务器上
/// </summary>
/// <param name="fileData">附件信息</param>
/// <param name="guid">附件组GUID</param>
/// <param name="folder">指定的上传目录</param>
/// <returns></returns>
[AcceptVerbs(HttpVerbs.Post)]
public ActionResult Upload(string guid, string folder)
Ł
   //如果需要修改字段显示,则参考下面代码处理
    dynamic obj = new ExpandoObject();
    List<string> urls = new List<string>();
    var result = new CommonResult();
    HttpFileCollectionBase files = HttpContext.Request.Files;
    if (files != null)
    {
       foreach (string key in files.Keys)
        Ł
           trv
           {
               #region MyRegion
               HttpPostedFileBase fileData = files[key];
               if (fileData != null)
```

案例对商品的详细信息的富文本进行编辑,来进行图片文字的编辑处理,如下界面所示。

基本信息	商品展示图片	商	品详细的	介绍		3)									
20	B U 🖉	Oper	Sans •	· 🖌	<mark>A</mark> -	:=	;≡ ≡ •	=	•	Ð		• 2	< <	:/>	?	
1	и <u>()</u> тч еа		51	00003		100003	阮漂均				135/55	09680				
			6 1	00004		100004	邓卉				139170	28898				
			7 1	00006		100006	首暁雪				186740	42588				
			8 1	00009		100009	王景景				137616	501760				
			9 1	00012		100012	朱丽华				136612	46623				
			10 1	00015		100015	土田 15				133/0/	1/219				
			12 1	00018		100018	物理把				139903	56055				
	46		13 1	00021		100021	當志超				186966	65275				
			14 1	00022		100022	周黎晶				134523	32225				
			15 1	00023		100023	邱海维				131023	21127				
	and the second															
1	一調整 上海分公司	- 11	16 1	00024		100024	张烨				186235	55060				
	- 222 上海分公司		16 1 17 1	00024		100024 100025	张烨 陈晓萌				186235	55060 100401				
	 □ 正海分公司 □	~	16 1 17 1 《 共 4	00024 00025 64 亲记 穿	录,每页	100024 100025 回 20 条,共	张烨 陈嘏萌 24 页]	<]	< 1	186235 186234	555060 H00401	当前页	ī (Ş	出全部词	• 5
代码 整 化码 握 接 速 整 级 、 纷 》 》 》 》 》 》 》 》 》 》 》 》 》 》 》 》 》 》		· · · · · · · · · · · · · ·	116 1 17 1 ま 4 第 4 第 4 第 4 第 4 第 5 9 8 8 4 10 17 1 17 1 17 1 1 17 1 1 17 1 1 17 1 1 17 1 1 17 1 1 1 17 1 1 17 1 1 1 17 1	00024 00025 64 余记家 友工 人工 与 在 框 外 で い の の の の の の の の の の の の の の の の の の	₽, ep E C R C R C C C C C C C C C C C C C	100024 100025 1 20条,共 主 古息,; Vinfor 工作,;	^{46,4} i Anna e2Sharp 把不同类 rm界面和 并能够迅) 是 灵 型 り し Wel 速 編	建業以下	< 1 , 它I 定 (件 加 , 1 五 行 起	186225 186234 3 3 3 4 3 4 4 5 3 4 5 4 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 7 7 8 7	555060 100401) [9世 不同的功速 人而3	当前前 的页生实 现	〕[\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	▶ 1
代码 整 程 读 题 下 文 、 《 ¹ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		、 、 、 、 、 た 、 た 、 た 、 た 、 た 、 た 、 た 、 た 、 、 、 、 、 、 、 、 、 、 、 、 、	16 1 17 1 ま 共 4 速 开 5 の 生 の 生 の を の で の で の で の で の で の で の で の の で の	00024 00025 64 条记录 发工具 全体框 分下框 分计于	₹, ∰ 具Dai 星 星 架 役 い て 大 こ こ 、 、 、 、 、 、 、 、 、 、 、 、 、	100024 100025 1 20条,共 吉息,: Vinfor 工作,:	^{##弾} Faitama 24页)是灵型的]Wel 速编	■ 魂 炎 界 译 過 ■	< Ⅰ ,它吃加, 一 一 一	186225 186234 3 3 3 4 3 4 4 5 3 4 5 4 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7 7 8 7	555060 100401 〕 <u>9</u> 世 不同 如 中 小 大 同 り 小 大 の り 、 の の の の の の の の の の の の の の の の の	的顶生现	到 舉 至架 (中 , , 地快返	上 4 年 1 年 1 年 1 年 1 年 1 年 1 年 1 年 1 年 1 年	

当我们插入图片的时候, 弹出一个对话框界面, 选择文件上传后返回 URL 显示在 SummerNote 插件上。

插入图5 抹信自 商	ī							
◎平旧恋 同。 从本地上传								
⊁→ B 选择文件	未选择任何	文件				1	?	
接口框架的图片地址								
有组件模块								

或者基于局								
					t	€入图止		
					35			
	_		_	_				
<u>Winform</u> 主体界面								
Winform主体界面		库存查询 - 广州委表	油软件科技有限公	司·深田之星仓库管理系			_	
Winform主体界面 ■ ■ -	疫管理 帮助	库存查询 - 广州委会	迪欽烨科技有限公	司-深田之星会库管理系			-	
Winform主体界面	·管理 帮助	库存查询 - 广州裂表	迪蚊件科技有限公 10				-	۵ ۵
Winform主体界面	建建 帮助	麻存査询 - 广州要表 10 92 92 2010日年三、 同志等用	油软件科技有限公 ● □□□□□●用 へ				-	۵ ۵
	於管理 就助	库存査询 - 广州製表 図 <u>9</u> 音件及成在号入 降時管理	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	司·派田之皇会库管理事	(統	_	-	â
Winform主体界面 19- 仓库管理 出分相表 系約 4世 中 一 章 《 条件入案 条件闭道 集存重要 条件原题	<立法 計算 一	<u>库存憲</u> 頃 - 广州最寄 区 全 省住 20座 存号 入	油飲件科技有限公 ■ ■ □	司·源田之皇会库管理事 大通讯录 公共通讯录 其伯管理	(統) 客户管理 。		-	۵ ۵
Winform主体界面 10- ① 金属電源 11分相差 系数 ・10- ① 金属電源 11分相差 系数 ・10- ② 金属電源 11分相差 系数 ・10- ③ 金属電源 ・10- ③ 金属電源 ・10	- 計算 (1) - 注意 - 注 - 注 - 注 - 注 - 注 - 注 - 注 - 注	库存查询 - 广州委委 2	油軟件科技有限公 ■ 見工管理 个) 5	司·凝田之星会库管理事 一 人通讯录 公共通讯录 其他管理	(統		-	۵
Winform主体界面 ● ● ● 金葉線 业务被表 系統 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	結算運 教助	底存查询 - 广州最表 10 91 92 省住恐迩存与入 函房管理 故语智厚	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	司·源田之堂会库管理事 人面识录 公共通讯录 其他管理	(統 繁) 客户管理 『	6	-	□
Winform主体界面 ● B- D- 在年間: 山方福美 新 → - ● ● ※行入庫 冬行記庫 医存置時 条件活用 - 合年間: ● - 合年間: ● - 一 ● - ● - 一 ● - 一 ● - 一 ● - ● - ● - □ - ● - ● - ● - ● - ● - ● - ● - ●	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	<u>库存壹词</u> - 广州最高 図 勉 當住及底存导入 底房管理 就請當得 — 當件葉表	這次并科技有限公 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	司 提用之量合库管理事 一 () 人面很示 () 其他管理 关別	(統 留户管理 下 一 則與商		-	۵ ۰
Winform主体界面 ・ 10- 0年 2日 2日 2日 2日 2日 2日 2日 2日 2日 2日		成存室词。广州裂表 区 12 當什及地存為人 進時管理 就該當時 一 曾件賞曲 一 曾件賞曲 一	油飲件科技有限公 ●	司·爱田之重合库管理事 入面由录 公共面讯录 其他管理 关别	4.统 ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	- x	-	
Winform主体界面 100 企業管理 出会報告 単価 100 企業管理 出会報告 単価 100 企業管理 単合 単価 100 企業管理 ● ● ● 100 企業管理 ●	# 第時 # 第時 初始化业务数据 5 存立曲 × 通年公庫 - 登件綱約 - 単位	広存室田・广州展示	這次件科技有限公 ◎ ◎ ○	司 2002 星合库管理师 入通讯录 公共通讯录 其他管理 关别	 (純) (新) (M) (M) (M) (M) (M) (M)	▼ 末 × ▼ 木 林 燕 ~ ───────────────────────────────────		
Winform主体界面 ● B- DP 公共管理 社会接美 単数 ・ 日本 公共管理 社会接美 単数 ・ 日本 公共管理 日本 一番 ● ・ 日本 (1)(所名音) ・ ● 日本 (1)(所名音) ● 日本 (1)(前名音) ● 日本 (1)(前名) ● 日本 (1)(前)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)	#當理 帮助 (子) 初始化业务数据 (古) (古) (古) (古) (古) (古) (古) (古) (古) (古)	<u>库存室</u> 道 - 广州最高	ibtyl并科技有限公 更工管理 个/ 页工管理 个/ 图号 单价	司 - 第田之堂 会库管理系 入面 () 泉 人面 () 名 人面 () 泉 人面 () 名 人面 () 名 () 名 () 名 () 名 () 名 () () 名 () () () () () () () ((统	▼ 来 ▼ 村商 ~		口 () () () () () () () () () () () () ()
Winform主体界面 ・ 10- 0年10- 0年10年 15日 15日 15日 15日 15日 15日 15日 15日	と言語	库疗查说 - 广州衰去	進放時科技有限公 第 の の の 、 の 、 、 、 、 、	司 深田之皇 全库管理系 通	(4) 家戸管理 夏 原本型号 原本型	- 未× - ▼ 村務 ~ 岸位		ロ 。 ・ で 注注性
Winform主体界面 ● ● <td></td> <td>広存室田・广州展示 全 全 全 た か に 市 の に か 一 、 の に で 一 、 の に で し の に つ に の に の に の に で し に に で で の し に に で で で し に に で で で し に に で で で で し に に で で で し に に で で で し に に で で で し に い に で で で し 、 に で で で し 、 に で で で し 、 の た つ し 、 の た つ し し い に つ し し い に つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ い こ つ し い こ つ し い こ つ し つ つ し つ し つ し つ し つ つ し つ つ し つ し つ つ し つ し つ つ し つ つ つ し つ つ つ つ つ つ つ つ つ つ つ つ つ</td> <td>進数與科技有限公 第 の工業環 へ の工業環 へ 等 単 新 の工業環 へ の工業 第 の工業環 へ の工業 の工業 の工業 の工業 の工業 の工業 の工業 の工業</td> <td>■.深田之量合库管理要 入通讯录 公共通讯录 其他管理 </td> <td> (統 家产管理 」 (別局商 」 (別局商 」 (別局商 」 (副局商) (副局商) (副局商) (副商商) ((III) (III) (IIII) (IIIII) (IIIIII) (IIIII) (IIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIIII) (IIIIIIIIII) (IIIIIIIIIIIIIIIII)</td> <td>- ★w - ★ - 東位 </td> <td></td> <td>口 。 。 · · · · ·</td>		広存室田・广州展示 全 全 全 た か に 市 の に か 一 、 の に で 一 、 の に で し の に つ に の に の に の に で し に に で で の し に に で で で し に に で で で し に に で で で で し に に で で で し に に で で で し に に で で で し に い に で で で し 、 に で で で し 、 に で で で し 、 の た つ し 、 の た つ し し い に つ し し い に つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ し い こ つ い こ つ し い こ つ し い こ つ し つ つ し つ し つ し つ し つ つ し つ つ し つ し つ つ し つ し つ つ し つ つ つ し つ つ つ つ つ つ つ つ つ つ つ つ つ	進数與科技有限公 第 の工業環 へ の工業環 へ 等 単 新 の工業環 へ の工業 第 の工業環 へ の工業 の工業 の工業 の工業 の工業 の工業 の工業 の工業	■.深田之量合库管理要 入通讯录 公共通讯录 其他管理 	 (統 家产管理 」 (別局商 」 (別局商 」 (別局商 」 (副局商) (副局商) (副局商) (副商商) ((III) (III) (IIII) (IIIII) (IIIIII) (IIIII) (IIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIII) (IIIIIII) (IIIIIIIIII) (IIIIIIIIIIIIIIIII)	- ★w - ★ - 東位 		口 。 。 · · · · ·
Winform主体界面 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	本語理 教助 びか役込券教授 「「「「」」」 「「」」 「「」」 「「」」 「」 「「」」 「」 「「」」 「」 「「」」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 「」 」 「」 」 「」 」 」 」 」 」 」 」 二		進次得終大者現公 第 「 」 の工業 の 「 」 「 」 「 」 「 」 「 」 」 「 」 「 」 」 、 」 、 」	 現日之星今年管理事 () ()	(株 第二世世 日 前知尚 月 市 中 田 二 月 市 全 第 二 世 一 日 二 日 二 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 一 日 二 二 二 二	· · · · · · · · · · · · · · · · · · ·		ロ の ・ 空永は 本 443
Winform主体界面 ● 18- 10- 位素管理 社会福祉 系彩 ● 10- 位素管理 社会福祉 系彩 ● 10- 位素管理 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●		応行登记 - 广州発表 ● 第4 天然在存名入 成時間標 ● 第4 天然在存名入 成時間標 ● 第4 天然 ● 第5 天 ● 第5 天然 ● 第5 天 ● 第5 王 ● 第5	進放得科技有限公 第 5 5 5 5 5 5 5 5 5 5 5 5 5	 現日之量会库管理事 通知法律 人計由:(示) 人計由:(示) 人計由:(示) (和行型) (和行型)<td>Skt 案户堂课 資料局商 資料局商 資料合売号 原件里 第件型 第件型 第1</td><td>· → → · → 利用 ~ 一 単元 ¥5,00.00 ¥228,031.20</td><td>世演 第 総格型号 NSK 22230CAMB NSK 22230CAMB NSK 22230CAMB</td><td>□ ○ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・</td>	Skt 案户堂课 資料局商 資料局商 資料合売号 原件里 第件型 第件型 第1	· → → · → 利用 ~ 一 単元 ¥5,00.00 ¥228,031.20	世演 第 総格型号 NSK 22230CAMB NSK 22230CAMB NSK 22230CAMB	□ ○ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
Winform主体界面 ● ●	2 素助 () () () () () () () () () () () () () (広存室田・广州委若 全 各作力採在符入、成房間便 ・ 各件支持 会件本計 ・ 体用位置 ・ 280204600109 6605 228020100103 6605 228021000103	油飲件科技有限公 第 の工業 の工業 の工業 の工業 の工業 の工業 の工業 の工業	时,深田之重合库管理要 人通讯录、公共通讯录 其他管理 学们	 (統) 第户管理 別局間 第合型 第合型 第合型 第合型 	▼ 未来 ~ 対称 ~ 厚心	<u>支</u> 適 素 契応型号 NKX 22320AM NKX 22320AM NKX NM-1-1900	□ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
Winform主体界面 ● B- DP 公共管理 社会接美 単純 学社入業 各件が進 正存立場 条件指標 会共管理 日本 金 条件入業 各件が進 正存立場 条件指標 一 条件方型電 保持入度 各件は定 正 新存至回 単素原目時間 ● 1年度(1)(指名) ● 2年度(2)三前) ● 3年度(2)三前) ● 3年度(2)=(2) ● 3年(2) ● 3年(2)=(2) ● 3年(2)=(2)=(2)=(2)=(2)=(2)=(2)=(2)=(2)=(2)=	法論理 款助 10年位39表別 5年支油 × 単位 単行 12 3 4		這款件科技者解公 意工業理 个」 页工業理 个」 页工業理 个」 例目記称 ▲ 報表 物表 物表 物表	 現日之星今年管理事 (第二) (第二) (第二) (第二) (第二) (第二) (第二) (第二)	4.46 第二世世世 第二世世世 第二世世世 第二世世 第二世	 ● 利用 ● 利用 ○ 単応 ● 単位 V6,300.00 V228,031.20 V12,461.40 V5,515.60 V	<u> 支加</u> 素 泉和型号 NSK 49230CAMB NSK 4913-CAMB NSK 4913-CAMB	□
Winform主体界面 ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ○ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	株式理 株助 初催化业务数据 50 年 空御 × 単位 単位 単位 単位 単位 単位 単位 ・ ・ ・ ・ ・ ・ ・ ・ ・	応行室地 - 广州発表 ● 第4 天然在存名入 原営理想 会社 次組在存名入 原営理想 会社 次組存名 ● 第4 天為 ● 第5 天 ● 第5 王 ●	油飲料料技有課公 (別) の工業課 へ) の工業課 へ) の工業課 へ) の工業課 へ 単一 単一 単一 単一 単一 の工業課 へ) の工業課 へ) の工業課 へ) の工業課 () の工業課 へ) の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業課 () の工業) () の の の の の の の の の の の の の の の の の の	司 源田之堂 会库管理事 通知 () 入田山泉 () 人田山泉 () 人田山泉 () 人田山泉 () 本() 本() 本() 本() 本() 本() 本() 本	Skit 第二世報	■ 基本 ● 材積 ~ 単位 ¥258,031.20 ¥228,031.20 ¥2,545.40 ¥4,991.40	武庫 第 派権型時 形容 222306/AME NSK 2413-06AM NSK 2	□ ○ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
Winform主体界面 100 公式協議 100 公式(100) 100 公式(100) 100 公式(100)	 株面接 数助 (1) (1)<!--</td--><td>広存室田・广州製茶 全 全 全 大部本市会入 成時間間 本 本 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市 本 本 本 本 本 本 本 本 本 本 本 本 本</td><td> 油飲料料技有限公 第 1 <li< td=""><td>■.深田之量会库管理要 人通讯录 公共通讯录 共制化管理 単化管理 マン、和文の マン、和文の マン、和文の マン、和文の マン、 マン、 マン、 マン、 マン、 マン、 マン、 マン、</td><td> ・ 計局局 ・ ・ ・</td><td>▼ 4月3 ~ 4月3 ~ 単元 ¥2,8,01.20 ¥12,461.40 ¥12,461.40 ¥12,451.960 ¥1,091.40</td><td>世面 ま</td><td>۰ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۰۰۲ ۲۰۰۲ ۲۰۰۲ ۲۰</td></li<></td>	広存室田・广州製茶 全 全 全 大部本市会入 成時間間 本 本 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市会入 本 市 大部本市 本 本 本 本 本 本 本 本 本 本 本 本 本	 油飲料料技有限公 第 1 <li< td=""><td>■.深田之量会库管理要 人通讯录 公共通讯录 共制化管理 単化管理 マン、和文の マン、和文の マン、和文の マン、和文の マン、 マン、 マン、 マン、 マン、 マン、 マン、 マン、</td><td> ・ 計局局 ・ ・ ・</td><td>▼ 4月3 ~ 4月3 ~ 単元 ¥2,8,01.20 ¥12,461.40 ¥12,461.40 ¥12,451.960 ¥1,091.40</td><td>世面 ま</td><td>۰ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۰۰۲ ۲۰۰۲ ۲۰۰۲ ۲۰</td></li<>	■.深田之量会库管理要 人通讯录 公共通讯录 共制化管理 単化管理 マン、和文の マン、和文の マン、和文の マン、和文の マン、 マン、 マン、 マン、 マン、 マン、 マン、 マン、	 ・ 計局局 ・ ・ ・	▼ 4月3 ~ 4月3 ~ 単元 ¥2,8,01.20 ¥12,461.40 ¥12,461.40 ¥12,451.960 ¥1,091.40	世面 ま	۰ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۵۲۲ ۲۰۰۲ ۲۰۰۲ ۲۰۰۲ ۲۰

上传文件成功后,通过下面的代码插入一个图片,如下代码。

```
$.ajax({
    data: data,
    type: "POST",
    url: "/FileUpload/Upload",
    cache: false,
    contentType: false,
    processData: false,
    success: function (json) {
        var data = $.parseJSON(json);
        var url = data.urls[0];
        $("#Note").summernote('insertImage', url, 'image name'); // the insertImage API
    });
```