《JavaFast 快速开发平台》 开发文档

	目录	
基本	5介绍	2
1.1.	前言	2
1.2.	平台介绍	3
1.3.	技术特点	3
1.4.	安全考虑	4
1.5.	技术选型	4
1.6.	安装部署	6
代码	9生成器	7
2.1.	操作步骤	7
	2.1.1 数据模型设计	8
	2.1.2 表单设计	9
	2.1.3 生成代码	
	2.1.4 生成菜单	12
	2.1.5 刷新代码工程,重启	12
	2.1.6 一对多注意事项	12
报表	電置器	13
3.1.	报表配置	14
常用	1组件	15
4.1.	用户工具 UserUtils	15
4.2.	缓存工具	15
4.3.	字典工具	15
4.4.	权限控制	16
4.5.	数据权限	16
4.6.	分页组件	17
	基本 1.1. 1.2. 1.3. 1.4. 1.5. 1.6. 代 2.1. 报 3.1. 常 4.1. 4.2. 4.3. 4.4. 4.5. 4.6.	基本介绍

	4.7.	树选择组件	17
	4.8.	多选下拉组件	17
	4.9.	部门选择组件	17
	4.10.	人员选择组件	
	4.11.	文件选择、文件上传组件	
	4.12.	图片选择、图片上传组件	
	4.13.	富文本在线编辑器组件	
	4.1.	工具类	
	4.2.	自定义标签	19
5.	系统酶	2置	19
	5.1.	基本配置	19
	4.	.1.1. 数据源配置	19
	4.	.1.2. 系统配置	20
	5.2.	阿里云短信配置	21
	5.3.	微信支付配置	21
6.	文件结	吉构	22
	6.1.	工程目录	22
	6.	.1.1 代码目录	22
	6.	.1.2 资源目录	23
	6.	.1.3 前端目录	23
7.	常见问	可题	24
	7.1.	Tomcat 中文乱码	24
	7.2.	Tomcat 内存溢出	24

1. 基本介绍

1.1. 前言

企业在项目开发的过程中,需要要写很多很多的代码,但仔细的分析来看,很多代码基本上都是重复的编写,功能大同小异。

快速开发平台作为一种新型的开发工具,它的出现减少了代码编写量,简化了开发过程, 缩短了开发周期,提高了开发效率,节省了开发成本。越来越多的开发者使用快速开发平台 来完成自己的开发工作。普通用户在接受简单培训后也能够使用快速开发平台 DIY 出自己 所需的各种管理软件。

1.2. 平台介绍

JavaFast 是一款基于代码生成器的智能快速开发平台,基于代码生成器的开发方式,可以将编码工作量减少 80%以上,为企业提高开发效率,为企业节省项目研发成本,减少开发周期。

JavaFast 基于流行的 JAVA 开源技术上构建,扩展容易,学习成本低,可快速构建企业级项目,即满足企业项目实现快速开发的需求。

1.3. 技术特点

- 采用主流的 SpringMVC + MyBatis + Apache Shiro + Jquery + Boostrap + Ehcache + Redis + Ztree 等基础架构。
- 代码生成器,通过在线配置实现一个表模型的增删改查导入导出等功能, 无需一行代码。支持单表、一对多、树结构,左树右表等多种数据模型, 根据表生成对应的 Entity,Service,Dao,Sql,Controller,JSP 等,增删改查导 入导出功能生成直接使用。
- 丰富的页面组件库:工具栏组件,列表组件,树形组件,弹窗组件,表 单校验,文件上传等,开发维护非常高效。
- 自动生成增删改通用SQL, Mybatis动态SQL可以根据条件智能生成SQL
 语句,支持多种匹配方式(全匹配/模糊查询/包含查询/不匹配查询)。
- 5. 强大的 Excel 导出导入工具封装,支持大数据量,注解定义,简单配置 即可实现。
- 6. 系统支持多种风格的自动切换,可满足企业对 UI 的美观要求。
- 7. 前端支持手机或平板访问,响应式样式。
- 8. 支持多数据源,支持配置文件里面配置多数据源。
- 9. 支持报表在线配置,通过简单配置 SQL 语句即可实现图形报表展示。
- 10. 常用的工具类封装,缓存、校验、字典、组织权限等。
- 操作权限控制精密细致,控制到行列级,表单字段级,按钮级,通过数据权限实现不同的人看不同数据。
- 12. 分层设计:(数据库层,数据访问层,业务逻辑层,展示层)层次清楚,

低耦合,各层必须通过接口才能接入并进行参数校验(如:在展示层不可直接操作数据库),保证数据操作的安全

- 适合分布式部署,云平台,移动平台开发,具有卓越的通用性、高效性、 平台移植性和安全性。
- 14. 经过了专业压力测试,保证后台数据的准确性和页面访问速度

1.4. 安全考虑

- 分层设计:(数据访问层,业务逻辑层,展示层)层次清楚,低耦合,各 层必须通过接口才能接入并进行参数校验(如:在展示层不可直接操作 数据库),保证数据操作的安全。
- 2、双重验证:用户表单提交双验证:包括服务器端验证及客户端验证,防止用户通过浏览器恶意修改(如不可写文本域、隐藏变量篡改、上传非法文件等)而跳过客户端验证操作数据库。
- 3、安全编码:用户表单提交所有数据,在服务器端都进行安全编码,防止 用户提交非法脚本及 SQL 注入获取敏感数据等,确保数据安全。
- 密码加密:登录用户密码进行 SHA1 散列加密,此加密方法是不可逆的。
 保证密文泄露后的安全问题。
- 5、访问验证:系统对所有管理端链接都进行用户身份权限验证,防止用户 直接通过 URL 进行未授权页面。
- 6、数据验证:对指定数据集权限进行过滤,3种数据权限可供选择(所有 权限,本部门,本人数据),可根据需要自由扩展。

1.5. 技术选型

1、后端框架

Spring Framework 4.0

MyBatis 3.2

Hibernate Validator 5.1

Apache Shiro 1.2

Spring Task

Alibaba Druid

Log4j

2、视图框架

Spring MVC 4.0

JSP

3、JS 框架

JQuery 2.1

Layer3.0

jQuery Select2 jQuery zTree

My97DatePicker

CKEcitor

4、前端样式 Bootstrap3.0

H+

- 5、缓存框架 Ehcache 2.6 Redis
- 6、工具类

Apache Commons Jackson Xstream Dozer POI

7、数据库支持

Mysql5.5+

Oracle

MSSQL Server

8、开发环境

JDK1.7+

Eclipse 3+

Maven3+

1.6. 安装部署

1. 将 JavaFast 工程以 Maven 项目导入到 Eclipse。



代码工程导入成功后,右键代码工程点击 Properties,在 Java Build Path 目录将 JavaSE 修改为 jdk1.7。

e filter text	Java Build Path	()-
Resource ^	(B. Courses Ch. Duris at Librarian As Order a	ad formed
Builders	Source Projects as Libraries 🗞 Order a	nd Export
Deployment Assembly	HARs and class folders on the build path.	
Java Build Path	> 🛋 JRE System Library [jdk1.7.0_60]	Add JARs
Java Code Style	> Maven Dependencies	
Java Compiler		Add External JARs
Java Editor		Add Variable
Javadoc Location		
JavaScript		Add Library
JSP Fragment		Add Class Folder
Maven		1288
Project Facets		Add External Class Folder
Project References		
Run/Debug Settings		Edit
Server		0
Service Policies		Remove
Targeted Kuntimes		Minoria IAD Ella
Task Repository		wigrate JAR File
lask lags		
Walidation		
> > > >		Apply

2. 设置数据源: src/main/resources/javafast.properties

```
#mysql database setting
jdbc.type=mysql
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/javafast?useUnicode=true&cha
racterEncoding=utf-8
jdbc.username=root
jdbc.password=sa
```

- 3. 创建数据库并初始化数据:运行 javafast.sql 数据文件导入数据。
- 4. 右键项目 Run as > Maven Build, 输入 tomcat:run, 运行项目:

<u>N</u> ar	me: javafa	st					
E	Main	💧 JR	(E) 🧬	Refresh	🦻 So	ource	Environment
Ba	ise director	y:					
\${	[project_loc	:jav	afast}				
							<u>W</u> orkspace
	G		+	-			
	<u>0</u> 0	ais:	tomc	attrun			
	<u>P</u> rofil	es:					

注:为了防止内存溢出建议在 VM arguments 中增加:

-Xms256m -Xmx256m -XX:MaxNewSize=256m -XX:MaxPermSize=256m .

ame: javafast		
🛾 Main 🛋 JRE 🔍 🧬 Refre	esh 💱 Source 🖾 Environment 🔲 Common	
Runtime JRE:		
○ Workspace default JRE	(jdk1.8.0_151)	
O Execution environment:	~	Environments
Alternate JRE:	jdk1.7.0_60 ~	Installed JREs.
VM arguments:		
-Xms256m -Xmx256m -XX	:MaxNewSize=256m -XX:MaxPermSize=256m	,
]

5. 访问工程: http://localhost:8080/javafast,超级管理员账号: sysadmin,密码: sysadmin

2. 代码生成器

代码生成器,支持多种数据模型,根据表生成对应的 Entity, Service, Dao, Action, JSP 等, 增删改查/排序/导出导入 Excel/权限控制/功能生成直接使用。

2.1. 操作步骤

- 1. 建立业务数据模型,设计表结构
- 2. 进入表单配置模块,添加业务表配置
- 3. 进入表单配置模块,生成代码
- 4. 进入表单配置模块,生成菜单和权限

2.1.1 数据模型设计

2.1.1.1 单表

单表模型设计,红框内为必须有的字段:

	A1984	4	test_ua	ta @java	iast (.) -	300						
	≡	🔁 新	建昌保	麻噌	另存为	•	添加字段 🔸	插入字段	● 冒删除字	段 🔑 主	键 🕈 上	眵 ↓ 下移
	字段	索引	外键	触发器	选项	注释	¥ SQL预	览				
	名						类型		长度	小数点	不是 null	
Þ	id						varchar		30	0	\checkmark	<u>/</u> 1
	name						varchar		50	0		
	sex						char		1	0		
	age						int		3	0		
	mobile	e					varchar		11	0		
	email						varchar		50	0		
	addre	ss					varchar		50	0		
	create	by					varchar		30	0	\checkmark	
	create	_name					varchar		30	0		
	create	_date					datetime		0	0	\checkmark	
	updat	e_by					varchar		30	0	\checkmark	
	updat	e_name					varchar		30	0		
	updat	e_date					datetime		0	0	\checkmark	
	remar	ks					varchar		50	0		
	del_fla	g					char		1	0	\checkmark	

2.1.1.2 一对多

一对多模型设计,红框内为必须有的字段:

	对象	7	test_data	_main	@javafas	: (.) 🍃	test_c	lata_child	@javafast	(.)	
	≡	🔁 新	健昌的	解存 喧	另存为	<mark>₀</mark> ∐添カ	加字段	→■插入割	字段 ● 書册間	除字段 🛛 🔑	主键 ↑
2	字段	索引	外键	触发器	选项	注释	SQL	预览			
	名			ġ.	趣			长度	小数点	不是 null	
▶	id			v	archar			30	0	\checkmark	<u> </u>
	no			v	archar			30	0		
	sale_ty	pe		c	har			1	0		
	amour	nt		d	ecimal			12	2	\checkmark	
	invoice	e_amt		d	ecimal			12	2		
	status			v	archar			1	0		
	deal_b	у		v	archar			30	0		
	deal_d	ate		d	atetime			0	0		
	create	by		v	archar			30	0	\checkmark	
	create	date		d	atetime			0	0	\checkmark	
	audit_k	бу		v	archar			30	0		
	audit_c	date		d	atetime			0	0		
	remarl	ks		v	archar			50	0		
	del_fla	g		c	har			1	0	\checkmark	
					-	Èŧ					

2.1.1.3 树结构

树结构模型设计,红框内为必须有的字段:

	对象	1	t_tree @j	avafast () - 表							
	≡	🔁 新	建 🔒 🦸	府陷	房存为	。」添	加字段 🕂	插入字段	● 删除字	段 🔑 主	键 🕈 上和	侈 ↓ 下移
1	字段	索引	外键	触发器	选项	注释	SQL 预览					
	名					类型	<u>l</u>		长度	小数点	不是 null	
۲	id					var	char		30	0	\checkmark	<u> </u>
	paren	t_id				var	char		30	0		
	paren	t_ids				var	char		1000	0		
	name					var	char		50	0	\checkmark	
	sort					int			11	0		
	create	e_by				var	char		30	0	\checkmark	
	create	_date				dat	etime		0	0	\checkmark	
	updat	e_by				var	char		30	0	\checkmark	
	updat	e_date				dat	etime		0	0	\checkmark	
	remar	rks				var	char		50	0		
	del_fla	ag				cha	r		1	0	\checkmark	

2.1.2 表单设计

2.1.2 第一步,选择表

表名: test_data:业务数据表	从数据库导入表			
		表名:	test_data:业务数据表	Ŧ
下一步返回			下一步 返回	

2.1.2.2 第二步, 配置表单

表单设计															
• 表名:	test_data		*说明:	业务数据表											
* 类名:	TestData		父表表名:	无	▼ 当前3	5外键:	无			▼ 如果	有父表,请指定	2表表名和外键			
表单明细															
列名	说明	物理类型	Java类型	Java属性名称	主键	可空	插入	编辑	列表	查询	查询匹配方式	显示表单	类型	字典类型	排序
id	编号	varchar2(64)	String *	id	~		~				=	单行文:	* 4		1
name	名称	nvarchar2(100)	String *	name		~	~	~	~	~	Like	单行文:	* 2		5
sex	性别	char(1)	String *	sex		•	•	•	•	~	=	●选按	H *	sex	6
del_flag	删除标记(0	char(1)	String *	delFlag			~				-	#选按	∄ *	del_flag	13
age	年龄	int(3)	Integer *	age		~	•	~				# 单行文:	* 2		40
mobile	手机号码	varchar(11)	String *	mobile		~	~	~			=	# 单行文:	* *		50
email	电子邮箱	varchar(50)	String *	email		~	~	~			-	● 単行文	* 4		60
address	联系地址	varchar(50)	String *	address		~	~	~			=	# 单行文:	* *		70
create_by	创建者	varchar2(64)	String *	createBy.id			~					● 単行文	* 4		80
create_name	创建者	varchar(30)	String *	createName		~	~	~			-	# 单行文:	* *		90
create_date	创建时间	timestamp(6)	Date *	createDate			~					* 日期选	¥ *		90
update_by	更新者	varchar2(64)	String *	updateBy.id			~	~			=	* 单行文:	* *		100
update_name	更新者	varchar(30)	String *	updateName		~	~	~			=	● 単行文	* 4		101
update_date	更新时间	timestamp(6)	Date *	updateDate			~	~	~		=	- 日期选	ş .		110
remarks	备注信息	nvarchar2(255)	String *	remarks		~	~	~	~			* 多行文:	* *		120

表名:物理表表名

说明:物理表表描述

类名: 生成表关联的实体类名称

父表表名:关联父表的表名,外键:当前表关联父表的主键,如果当前 表为子表,需在此指定父表及外键。外键字段需在字段列表中手动设置 属性名(对象.主键,例如:将 userId 修改为 user.id)

列名: 数据表定义的字段名称

说明:数据表定义的字段注释

物理类型:数据表定义字段类型

Java 类型:实体对象的属性字段类型

Java 属性名称:实体对象的属性字段(对象名.属性名|属性名 2|属性名 3, 例如:用户 user.id|name|loginName,属性名 2 和属性名 3 为 Join 时关联 查询的字段)

主键: 是否是主键字段

可空: 该字段是否可为空

插入:是否是插入字段,如果是则包含在 insert 语句里

编辑:是否是编辑字段,如果是则包含在 update 语句里 列表:是否是列表查询,如果是则包含在列表页的表格列里。 查询:是否是查询字段,如果是则包含在查询页的查询列表里。 查询方式:查询字段的查询方式,也就是 where 后的条件表达式,如: 字段 1=字段 2 AND 字段 3>字段 4 AND 字段 5 !=字段 6。 字段生成方案:表单中字段生成的样式,如: input, select, treeselect, areatext 等等

字典类型:如果字段生成方案为:下拉框、复选框、单选框,则该字段 必须指定一个字典类型,字典类型为字段管理中的字典类型。 排序:字段生成的先后顺序,升序。

2.1.3 生成代码

生成		
* 业务表名:	test_data:业务数据表 🔹	生成的数据表,一对多情况下请选择主表。
*模板分类:	増删改査(単表) ▼	生成结构:{包名}/{模块名}/{分层(dao,entity,service,web)}/{子模块名//[java类}
* 生成包路径:	com.javafast.modules	建议模块包:com.javafast.modules
* 生成模块名:	test	可理解为子系统名,例如 sys
* 生成功能描述:	单表生成	将设置到判描述
* 生成功能名称:	单表	用作功能提示,如:保存"某某"成功
* 生成功能作者:	javafast	功能开发者
* 表单模型:	弾窗 ▼	编辑页面的显示方式
生成选项:	是否替换现有文件	

业务表名:要生成代码的业务表。

模板分类: 生成的模板,目前可生成如下模板: 增删改查(单表)、增删 改查(一对多)、仅持久层(dao/entity/mapper)、树结构表(一体)。 生成包路径: 生成哪个包下。 生成模块名: 生成包下的模块名称,模块名称下进行分层。 生成功描述: 生成到类注释里。 生成功能名: 生成功能提示,如列表上、提示信息等。 生成功能作者: 开发者姓名 生成选项: 是否替换现有文件,提供重复生成,覆盖原有文件。

2.1.4 生成菜单



也可以根据控制器@RequestMapping 及@RequiresPermissions 手动添加菜单和权限标志

2.1.5 刷新代码工程,重启

刷新代码,检查没有错误后,重启工程,查看生成的页面效果。

2.1.6 一对多注意事项

一对多生成需要配置一个主表,和一个或多个子表。子表配置需要指定父表 表名和外键,如下:

	* 表名:	test_data_child				*说明:	订单	包明细(—)	対多子表))						
	* 类名:	TestDataChild				父表表名:	test	_data_ma	ain : 订单	(*	当前表外	罐: ord	er_ld:所属订单	▼ 如果有父表,请指;;	E父表表名和外键	
表单明细					l											
列名	说明	物理类型	Java类型		Java腐性名称		主键	可空	插入	编辑	列农	查询	查询匹配方式	显示表单类型	字典类型	排序
id	编号	varchar(30)	String	٠	id		~		•				= *	单行文本 *		10
order_id	所属订单	varchar(30)	自定义	*	order.id	请用XX.id		•	•	•			= *	单行文本 *		20
product_id	商品	varchar(30)	自定义	٠	product.id]name				~	~	~			单行文本 *		30
unit_type	单位	varchar(30)	String	Ŧ	unitType			~	•	~	~		= *	单行文本 *		40
orice	单价(元)	decimal(12,2)	BigDecimal	Ŧ	price				•	~	~			单行文本 🔻		50
num	数量	int(11)	Integer	Ŧ	num				•	~	•		-	单行文本 🔻		60
mount	金额(元)	decimal(12,2)	BigDecimal	٠	amount				•	~	~		-	单行文本 *		70
emarks	备注	varchar(50)	String	٣	remarks			•	•	~	~		= *	单行文本 *		80
lel_flag	删除标记	char(1)	String	Ŧ	delFlag				~				= ×	单选按钮 ▼	del_flag	90

表配置结果,如下:

业务表单列	业务表单列表								
表名: test_data 说明: 父表表名:									
+创建表的	+ 创建表单 C 刷新								
	表名 \$	说明	类名◆	父表⇔	操作				
	test_data_child	订单明细(一对多子表)	TestDataChild	test_data_main	☞ 修改 ● 删除 ☞ 生成代码 ■ 生成菜单				
	test_data_main	订单(一对多)	TestDataMain		☞ 修改 ● 删除 ☞ 生成代码 ☞ 生成菜单				
显示第 1 到	第 2 条记录,总共 2 条记录 每页显示	10 🔺 条记录							

生成代码的时候,只需要选择主表进行代码生成。

在代码生成页面选择"增删改查 (一对多)",业务表表名选择"主表",如

下:

* 业务表名:	test_data_main : 订单(一)	▼ 生成的数据表 , 一对多情况下请选择主表。
* 模板分类:	增删改查(一对多)	▼ 主成结构:{包名}/{模块名}/{分层(dao,entity,service,web))/{子模块名}/{java类}
* 生成包路径:	com.javafast.modules	建议模块包:com.javafast.modules
* 生成模块名:	test	可理解为子系统名,例如 sys
* 生成功能描述:	订单信息	将设置到美描述
* 生成功能名称:	订单信息	用作功能提示,如:保存'某某'成功
* 生成功能作者:	javafast	功能开发者
*表单模型:	页面	▼ 编辑页面的显示方式
生成选项:	是否替换现有文件	

其它操作与单表生成相同。

3. 报表配置器

在线配置报表,无需编码,一分钟快速生成自定义图形报表(无需编码,通过在线配置 方式,实现曲线图,柱状图,数据等报表)。 3.1. 报表配置

·≡ JAVAFAST		主页	(15	響户	财务	仓储	简介	购买	官网	₩\$
总经理	×	图表	配置列表							
0 0 ø		名称				表名:				图表类型: ▼
□ 代码生成	×	+	āta 🗟	修改	自删除	C周新				
表单配置			2	課 9 (1) (1) (1)			描述	• hong8		
图表配置			周	存分布图			库存	分布图		
卤 生成示例	>		R	品销量比	例分析		商品	销量比例	份析	
∠ 统计报表	>		¥	浩 阶段占	比分析		销售	机会里面	i的各个阶	级的比例分析报表
▲ 客户管理	>	显示	第 1 到第	3 条记录	, 总共 3 {	条记录 每了	显示	10 - 祭	记录	
□ 进销存管理	>									

报表支持柱状图、饼状图、线壮图,配置报表查询所需要的 SQL 语句,以及报表显示的相关字段。配置图表请确保 SQL 语句的正确性。

Bit ##797-0000 Mile ##797-0000 Bit ##797-0000 Mile ##797-0000 Bit Mile ##797-0000 Mile ##797-0000 Bit Mile ##797-0000 Mile ##797-0000 Mile # Bit Mile ##797-0000 Mile ##797-0000 Mile #		修改图表配置								- 2 ×
86 87.5: ums_stock 87.8.72: 97.002 • 76 X4970: name • Y4970: count • 76 Aller Count • Y4970: count • 76 Aller Count • State Count • • 76 Aller Count • • • • • 78 Aller Count • • • • • • 78 Aller Count • • • • • • 78 Aller Count • • • • • • 78 Aller Count • • • • • • • • 788 •	10 101	名称:	库存分布图	•		1983년 :	库存分布图			
Mit/Pill: name · YM/P/R: count · Mit Material: ECOT movement system, unit) Mit count FROM units, stock is · <	88	表名:	wms_stock			問表类型:	饼状图	* *		
Res SELECT wind wardene sum(s stock, num) AS court FROM wing, stock is LEFT_ON wing, wardhouse_ of a wardhouse_ of group Br is wardhouse_ of selections SELECT wind wardene sum(s stock, num) AS court FROM wing, stock is LEFT_ON wing, wardhouse_ of group Br is wardhouse_ of group Br is wardhouse_ of selections REMEMONIC SELECT wind wardhouse_ of Br is wardhouse_ of group Br is wardhouse_ of group Br is wardhouse_ of selections SELECT wind wardhouse_ of Br is wardhouse_ of Br i	库车	X铀字段:	name			YM49472 :	count			
Bits State State <ths< td=""><td>198</td><td></td><td>SELECT w.no,w.name</td><td>e,sum(s.stock_num) AS c</td><td>ount FROM wms_st</td><td>ock s</td><td></td><td></td><td></td><td></td></ths<>	198		SELECT w.no,w.name	e,sum(s.stock_num) AS c	ount FROM wms_st	ock s				
1988 Max2000 PRAXE000 PRAXE000 PRAXE0000 PRAXE000 PRAXE0000 PRAXE0000 PRAXE00000 PRAXE0000 PRAXE00000000 PRAXE000000000000000000000000000000000000	893	查询数据SQL:	LEFT JOIN wms_ware GROUP BY s.wareho	ehouse w ON w.id=s.warei use_id	house_id					
REACEBRE PERM PERM PERM REAL	1 到銅 3									
9535 95388 95388 95388 8659 8659 8677 8677 9687		國表配置明細								
nume 10年8時 学研書 2 <th2< th=""> 2 <th2< td=""><td></td><td>字段名</td><td>字段描述</td><td>字段类型</td><td>是否显示</td><td>sdn.net/shiyiho 最否查询</td><td>查询方式</td><td>显示类型</td><td>字典类型</td><td>建床 (</td></th2<></th2<>		字段名	字段描述	字段类型	是否显示	sdn.net/shiyiho 最否查询	查询方式	显示类型	字典类型	建床 (
No O P max P max<		name	仓库名称	字符串 ~	是	· ·	~	~		
Count 254772 5532 V 2 V V V		no	仓库编号	字符串 >	A	· ·	~	~		
		count	总库存量		*	· ·	~	~		
		新聞								

点击保存,图表配置完毕,可测试查看效果

memory and 2345.com	- pavatast	Ers			- 0.4.191	100			a secondaria			
JAVAFAS1		主页任务 第	e NS	合國 副介	88	nH ES			20 B B B	A 1832 🛈	8 Mit 9	\$1576-
总经理 00 #		库存分布图 5-90-05					-					1
2 代码在18	8											
图示配限												
) ±12791												
4. 杭计探索												
有户管理		🕳 4800 🦔 102	60				c net/ahiy thos	108 - 589				
1 进转存管理												
目 口和描述		库存分布图列表										n # #
] 财务管理		30(9)										
TRANSFER		仓库各带				仓库编号			总库存量			
2 OADS		16220				001			11			
■ KR11分析		中南台				002			100			
系统设置												

4. 常用组件

4.1. 用户工具 UserUtils

应用场景:在 java 文件或 jsp 页面上,获取当前用户相关信息

 获取当前用户:

 UserUtils.getUser();

 \${fns:getUser()}

 获取当前部门:

 UserUtils.getOfficeList();

 \${fns:getOfficeList()}

 获取当前用户有权限的菜单:

 UserUtils.getOfficeList();

 \${fns:getOfficeList()}

 获取当前用户有权限的菜单:

 UserUtils.getOfficeList();

 \${fns:getOfficeList();

 \${fns:getOfficeList();

UserUtils.getCache(key);

\${fns:getCache(cacheName, defaultValue)}

设置当前用户缓存:

UserUtils.putCache(key);

4.2. 缓存工具

应用场景:系统字典

- 1. 设置应用程序缓存: CacheUtils.put(key);
- 2. 获取应用程序缓存: CacheUtils.get(key);

4.3. 字典工具

应用场景:系统全局固定的字典数据, java 或 jsp 中获取字典相关数据。

3. 根据类型和值获取字典标签 (列表取值):

DictUtils.getDictLabel(String value, String type, String defaultValue)

\${ fns:getDictLabel (value, type, defaultValue)}

4. 根据类型和标签获取字典值(根据标签取值):

DictUtils.getDictValue(String label, String type, String defaultLabel)
\${fns:getDictValue(label, type, defaultValue)}

5. 根据类型获取字典列表 (下拉框,复选框,单选框):

DictUtils.getDictList(String type)

\${fns:getDictList(type)}

4.4. 权限控制

应用场景:访问功能授权,查看权限,编辑权限,导入权限,审核权限。

1. 给方法添加权限标志

@RequiresPermissions("sys:user:view")

- 2. 菜单中设置权限标志。
- 3. 判断权限:

SecurityUtils.getSubject().isPermitted("sys:user:edit");

4. 视图中控制按钮 (shiro.tld):

<shiro:hasPermission name="sys:user:edit">

<input id="btnSubmit" class="btn btn-primary" type="submit" value="保存"/></shiro:hasPermission>

<!-- 任何一个符合条件的权限 -->

<shiro:hasAnyPermissions name="sys:user:view, sys:user:edit,">

<input id="btnSubmit" class="btn btn-primary" type="submit" value="返 回"/> </shiro: hasAnyPermissions>

4.5. 数据权限

应用场景:某用户访问数据范围:所有数据,部门数据,个人数据 1.在 Service 层需加入权限过滤方法: // 生成数据权限过滤条件 dataScopeFilter(entity);

```
2.在SQL 需加入${sqlMap.dsf}如:
<!-- 分页查询用户信息 -->
<select id="findList" parameterType="User" resultMap="User">
```

```
SELECT
<include refid="userColumns"/>
FROM sys_user a
<include refid="userJoins"/>
WHERE a.del_flag = '0'
<!-- 数据范围过滤 -->
${sqlMap.dsf}
</select>
```

4.6. 分页组件

// 设置分页参数,则分页,如果不设置,则根据条件获取全部
entity.setPage(page);
// 执行分页查询
page.setList(entityDao.findPage(entity));

4.7. 树选择组件

```
标签文件: /javafast/src/main/webapp/WEB-INF/tags/treeselect.tag
使用方法如:
<sys:treeselect id="company" name="company.id"
value="${user.company.id}" labelName="company.name"
labelValue="${user.company.name}" title="公司"
url="/sys/office/treeData?type=1" cssClass="form-control required"/>
```

4.8. 多选下拉组件

<form:select path="status" cssClass="form-control input-xlarge" multiple="multiple"> <form:options items="\${fns:getDictList(audit_status)}" itemLabel="label" itemValue="value" htmlEscape="false"/>

```
</form:select>
```

'audit_status '为数据字典类型

4.9. 部门选择组件

<tags:treeselect id="office" name="office.id" value="\${user.office.id}" labelName="office.name" labelValue="\${user.office.name}" title="部门" url="/sys/office/treeData?type=2" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/>

多选需要加 checked="true" 属性

4.10. 人员选择组件

<tags:treeselect id="user" name="user.id" value="\${user.id}" labelName="user.name" labelValue="\${user.name}" title="用户" url="/sys/office/treeData?type=3" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/> 多选需要加 checked="true" 属性

4.11. 文件选择、文件上传组件

<form:hidden path="name" htmlEscape="false" maxlength="255" class="input-xlarge"/>

<tags:ckfinder input="name" type="files" uploadPath="/test "/>

4.12. 图片选择、图片上传组件

<form:hidden path="name" htmlEscape="false" maxlength="255" class="input-xlarge"/>

<tags:ckfinder input="name" type="images" uploadPath="/test "/>

4.13. 富文本在线编辑器组件

<form:textarea id="name" htmlEscape="true" path="name" rows="4" maxlength="200" class="input-xxlarge"/>

<tags:ckeditor replace="name" uploadPath="/test " />

4.1. 工具类

UserUtils: 用户工具类(获取当前用户的相关信息)。

CacheUtils: Cache 缓存工具类。

CookieUtils: Cookie 操作工具类

DateUtils: 日期时间工具类。

FileUtils: 文件操作工具类。

StringUtils: 字符串操作工具类。

ExcelExcel 导入导出,参考用户导入导出例子。

Reflections: Java 对象操作反射工具类。

BeanMapper: Bean 与 Bean, Bean 与 Conllection 的互转 JaxbMapper: XML 与 Object 互转。 JsonMapper: JSON 与 Object 互转。 Cryptos、Digests: 密钥工具类 SHA1、MD5。 Collections3: 集合对象工具类。 Encodes: 各种编码转换工具类。 Exceptions: 异常工具类。 FreeMarkers: FreeMarkers 模板工具类。 Identities: 唯一标识生成算法工具类(uuid、random) PropertiesLoader: 属性文件操作工具类 Threads: 线程相关操作工具类。

4.2. 自定义标签

ckeditor.tag: HTML 在线编辑器。
ckfinder.tag: 在线文件管理。
iconselect.tag: Icon 图标选择。
message.tag: 消息弹框。
validateCode.tag: 验证码。

5.系统配置

5.1. 基本配置

javafast.properties 是系统的配置文件。

4.1.1. 数据源配置

#mysql数据库连接设置。
jdbc.type=mysql
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/javafast?useUnicode=true&charact

```
erEncoding=utf-8
```

jdbc.username=root
jdbc.password=123456

#oracle数据库连接设置
#jdbc.type=oracle
#jdbc.driver=oracle.jdbc.driver.OracleDriver
#jdbc.url=jdbc:oracle:thin:@127.0.0.1:1521:javafast
#jdbc.username=root
#jdbc.password=123456

```
#mssql 数据库连接设置
#jdbc.type=mssql
#jdbc.driver=net.sourceforge.jtds.jdbc.Driver
#jdbc.url=jdbc:jtds:sqlserver://localhost:1433/javafast
#jdbc.username=sa
#jdbc.password=sa
```

```
# 连接池设置,初始大小,最小,最大连接数。
jdbc.pool.init=1
jdbc.pool.minIdle=3
jdbc.pool.maxActive=20
```

```
# 测试连接sql语句
jdbc.testSql=SELECT 'x' FROM DUAL
```

4.1.2. 系统配置

```
# 配置产品名称,版权日期和版本号
productName=JavaFast 快速开发平台
copyrightYear=2017
version=V1.0
```

是否是演示模式,演示模式下部分模块无法进行修改数据的操作 demoMode=false

管理端跟路径 adminPath=/jf

前端跟路径 frontPath=/f

分页大小,默认每页10条 page.pageSize=10

```
# 设置通知间隔访问时间,单位毫秒.
oa.notify.remind.interval=60000
# 设置SESSION超时时间,web.xml里设置无效,单位毫秒.
session.sessionTimeout=120000
session.sessionTimeoutClean=120000
```

5.2. 阿里云短信配置



5.3. 微信支付配置

1. 登录微信公众号平台,并开通微信支付: <u>https://mp.weixin.qq.com/</u> 按照微信支付开通流程填写信息,并成功开通微信支付后,进入微信支付商户平台。 设置 API 秘钥,支付开发的时候会用到。



在微信公众平台-开发-基本配置,可以看到 appid

2. 在项目工程的 WxPayConfig.java 文件配置微信支付参数:



3. 支付测试:

完璧订单列	×.						
订单编号:		STIRIERS:	* 支付規型:	• 05829193:	m) -	m	
+1520	CRM						
	វេយអេមិ ៖					6938人 0	操作
	18032216203501	6000es			×	管理员	● 微信支付 ¥支付室支付
	18032117384501	订单编程: 10032 高可约用: 在107	216203501 订单金额: 0.01			10 HE (5)	
慶示館 1 到	Mi 2 条记录,总共 2 条记录:	梅页显示					
		S 网络麦村			支付金额: 0.01元		
			15MAR, HENDROGALLON	 若有利用用用 			
					完成支付 关闭		

6. 文件结构

6.1. 工程目录

6.1.1 代码目录

src\main\java

com.javafast	总目录
common	公共模块存放目录

beanvalidator	实体 Bean 验证相关类
log	日志工具相关类
mapper	各种 Object 到 Xml、Object 到 Json 的映射转换类
persistence	持久层相关类
security	安全相关类
service	业务层相关类
servlet	公共 servlet 相关类
utils	各种操作小工具类
web	模型控制器层相关类
modules	JavaFast 内置功能模块存放目录
sys	系统核心模块存放目录
dao	数据访问层相关类
entity	实体相关类
interceptor	拦截器相关类
service	业务处相关类
web	模型控制器层相关类
utils	系统模块的工具类

6.1.2 资源目录

src\main\resources

act	Activiti 工作流引擎相关文件(部署文件、bpmn)
cache	Ehcache 缓存配置存放目录
mappings	Mybatis Sql 映射文件存放目录
javafast.properties	系统配置属性文件
spring-*.xml	Spring 相关文件
log4j.properties	Log4j 日志配置属性文件

6.1.3 前端目录

src\main\webapp

static	静态文件存放目录(JS、CSS、前端插件类库等)
userfiles	用户上传文件目录
WEB-INF	WEB 应用安全目录,通过映射访问相关文件。
lib	依赖 jar 包目录
tags	Tags 标签存放目录
views	视图文件目录
error	系统异常映射相关页面
include	视图相关包含文件
layouts	视图布局相关文件
modules	内置核心功能模块视图相关文件
sys	系统管理模块视图相关文件

test	示例模块视图相关文件
ckfinder.xml	CKfinder 配置文件
decorators.xml	Decorator 配置文件
web.xml	Web 配置文件

7. 常见问题

7.1. Tomcat 中文乱码

修改 Tomcat 的 server.xml 文件的 Connector 项, 增加 URIEncoding="UTF-8"

7.2. Tomcat 内存溢出

java.lang.OutOfMemoryError: PermGen space

解决方案:

Eclipse 中以 Run as 运行项目, 在 VM arguments 中增加: -Xms256m -Xmx256m -XX:MaxNewSize=256m -XX:MaxPermSize=256m。

Name: javafast		
📄 Main 🛋 JRE 🛛 🤣 Refre	sh 🦆 Source 📧 Environment 🔲 Common	
Runtime JRE:		
○ Workspace default JRE (jdk1.8.0_151)	
O Execution environment:	~	Environments
Alternate JRE:	jdk1.7.0_60 ~	Installed JREs
VM arguments:		7
-Xms256m -Xmx256m -XX	:MaxNewSize=256m -XX:MaxPermSize=256m	^
		-
		×
		Variables

在 tomcat\bin 目录下找到 bin \ catalina.bat 文件,用 编辑形式打开,找到以下的内容:加入下边红色的 Tomcat 内存设置语句就可以了: rem ----- Execute The Requested Command

1. set JAVA_OPTS=-Xms1024m -Xmx4096m -XX:PermSize=128m -XX:MaxPermSize=256m

复制代码

<u>Linux</u>下 bin/ catalina.sh

rem ----- Execute The Requested Command

 CATALINA_OPTS="-Xms2048m -Xmx2048m -XX:PermSize=512M -XX:MaxNewSize=256m -XX:MaxPermSize=1024m"

复制代码