

技术文档

一、 技术架构

1. 基础框架 jeecg-p3 微服务开发框架
 - ① Java、JavaScript
 - ② Spring 、SpringMVC 、Mybatis、Mysql
 - ③ HTML、CSS、EChart、Vue
2. 表结构

商品表	f_d_goods
订单表	f_d_order
仓库表	f_d_store
仓库商品表	f_d_store_items
系统表	
	jp_system_auth
	jp_system_role
	jp_system_role_auth_rel
	jp_system_user
	jp_system_user_role_rel

二、 功能实现

1.前台 Model View Controller 入口：

/jeecg-p3-biz-fresh/src/main/java/com/fresh/main/IndexController.java

2.查询数据调用接口：

/jeecg-p3-biz-fresh/src/main/java/com/fresh/core/web/FDOrderController.java

/jeecg-p3-biz-fresh/src/main/java/com/fresh/core/web/FDStoreItemsController.java

/jeecg-p3-biz-fresh/src/main/java/com/fresh/core/web/FDStoreController.java

/jeecg-p3-biz-fresh/src/main/java/com/fresh/core/web/FDGoodsController.java

3.关键 DAO 层：

/jeecg-p3-biz-fresh/src/main/java/com/fresh/core/dao/FDQueryDao.java

4.登录、注销、权限：

系统管理菜单下可以增加删除菜单及子菜单、可以增加和修改用户并对用户的菜单访问权限进行设置。

5.各功能实现细节:

1) 仓库管理

```
@RequestMapping(params = "storage",method =
{RequestMethod.GET,RequestMethod.POST})

    public void storage(HttpServletRequest
request,HttpServletRequest response,ModelMap model)
throws Exception{

    try {

        VelocityContext velocityContext = new
VelocityContext();

        String viewName = "fresh/main/storage.vm";

ViewVelocity.view(request,response,viewName,veloci
tyContext);

    } catch (Exception e) {

        e.printStackTrace();

    }

}
```

数据查询代码:

```
@Controller

@RequestMapping("/fresh/fDStoreItems")

    public class FDStoreItemsController
```

.....

/**

* Find store grid data

*

* @return

*/

@RequestMapping(params = "findGrid", method =
{RequestMethod.**GET**, RequestMethod.**POST**})

@ResponseBody

public AjaxJson findGrid(@ModelAttribute DateVo
dateVo, @ModelAttribute FDGoodsEntity goods,
@ModelAttribute FDStoreEntity store,
@ModelAttribute FDStoreItemsEntity fDStoreItems) {
 AjaxJson j = **new** AjaxJson();
 LOG.info("{} ", dateVo);
 LOG.info("{} ", goods);
 LOG.info("{} ", store);
 LOG.info("{} ", fDStoreItems);
 try {
 List<StoreVo> list = fDQueryDao.getAll(goods,
dateVo, store);
 j.setObj(list);

```
        j.setMsg("保存成功");
    } catch (Exception e) {
        j.setSuccess(false);
        j.setMsg("保存失败");
        e.printStackTrace();
    }

    return j;
}
```

View:

/jeecg-p3-starter-web/src/main/resources/content/refresh/main/storage.vm

关键 JavaScript 代码:

```
<script>

    var labelOption = {
normal : {
        show : true,
        position : 'insideBottom',
        rotate:45,
        distance:10
    }
};
```

```
var vm = new Vue({
  el : '#app',
  data : {
    tableHead : ["gname", "gtype", "location",
"name", "type", "date",
    "count" ],
    tableLabel : {
      "id" : "ID",
      "gname" : "商品名称",
      "gtype" : "商品类型",
      "location" : "仓库地点",
      "name" : "仓库名称",
      "type" : "仓库类型",
      "date" : "入库日期",
      "count" : "商品数量"
    },
    tableData : [],
    myChart :
echarts.init(document.getElementById('main')),
    option : {
      legend : {},
      tooltip : {
```

```
        trigger : 'axis',
        axisPointer : {
            type : 'shadow'
        }
    },
    dataset : {
        dimensions : [ 'date', '蔬菜', '肉品', '水果', '水产', '干货' ],
        source : []
    },
    xAxis : {
        type : 'category'
    },
    yAxis : {},
    // Declare several bar series, each will be
    mapped to a column of dataset.source by default.
    series : [ {
        type : 'bar',
        label:labelOption
    }, {
        type : 'bar',
        label:labelOption
    } ]
}
```

```

    }, {
        type : 'bar',
        label:labelOption
    }, {
        type : 'bar',
        label:labelOption
    }, {
        type : 'bar',
        label:labelOption
    } ]
    }
},
created : function() {
    this.initChart()
},
methods : {
    initChart : function() {
        this.myChart.setOption(this.option, true);
    },
    flushChart : function() {
        var buf = {}
        for (var i = 0; i < this.tableData.length; i++)

```

```

{
    var o = this.tableData[i]
    // console.log(o.date, o.gtype, o.count)
    if (buf[o.date] == null) {
buf[o.date] = {}
    }
    if (buf[o.date].date == null) {
buf[o.date].date = o.date
    }
    if (buf[o.date][o.gtype] == null) {
buf[o.date][o.gtype] = 0
    }
    buf[o.date][o.gtype] +=
parseInt(o.count)
}
var source = []
for ( var e in buf) {
    source.push(buf[e])
}
this.option.dataset.source = source
this.myChart.setOption(this.option, true);
}

```



```

    }
    })
    $("#ckAll").click(function() {
    if ($("#this").prop("checked")) {
        $("#checkbox").prop("checked", true)
    } else {
        $("#checkbox").prop("checked", false)
    }
    });

    //jquery 获取复选框值
    function getCkValue() {
    var chk_value = [];
    $('input[name="ck"]:checked').each(function()
    {
        chk_value.push($("#this").val());
    });
    //alert(chk_value.length==0 ?'你还没有选择任何
    内容!':chk_value);
    }

    //获得所有选中行的 ID 进行批量删除
    function batchDelete() {

```

```
var ids = [];
var check = document.getElementsByName("ck");
for (var i = 0; i < check.length; i++) {
    if (check[i].checked) {
        ids.push(check[i].value);
    }
}
if (ids.length > 0) {
    parent.layer.confirm('确认进行批量删除吗?', {
        btn : [ '确定', '取消' ]
        //按钮
    }, function() {
        jQuery.ajax({
            url :
'$!{basePath}/fresh/fDStoreItems.do?batchDelete
',
            data : {
                ids : ids
            },
            type : 'POST',
            dataType : 'json',
```

```
        traditional : true,
        success : function(data) {
            if (data.success) {
                parent.layer.alert(data.msg, {
                    icon : 1,
                    shadeClose : false,
                    title : '提示'
                }, function(index) {

                    document.getElementById('formSubmit').submit
                    ();

                    parent.layer.close(index);
                });
            } else {
                layeralert(data.msg, 0);
            }
        },
        error : function(data, status, e) {
            if (data.status == "401") {
                layeralert("没有权限! ", 0);
                return;
            }
        }
    }
}
```

```

        }
    });
    });
} else {
    layeralert("您还没有选择要删除的行");
}
}

function findGrid() {
var inputs =
$("#formSubmit").find("input,select")
var postData = {}
for (var i = 0; i < inputs.length; i++) {
    var e = inputs.eq(i)
    var name = e.attr("name")
    var value = e.val().trim()
    if (value !== undefined && value !== null &&
value !== '') {
        postData[name] = value
    }
}
jQuery.ajax({
    url :

```

```
'${basePath}/fresh/fDStoreItems.do?findGrid',  
    data : postData,  
    type : 'POST',  
    dataType : 'json',  
    traditional : true,  
    success : function(data) {  
        if (data.success) {  
            vm.tableData = data.obj  
            vm.flushChart()  
        } else {  
            layeralert(data.msg, 0);  
        }  
    },  
    error : function(data, status, e) {  
        if (data.status == "401") {  
            layeralert("没有权限!", 0);  
            return;  
        }  
    }  
});  
  
}
```

//重置按钮

```
function searchReset() {
$("#formSubmit").find(":input").val("");
$("#formSubmit").submit();
}

function expPng() {
var img = new Image();
img.src = vm.myChart.getDataURL({
    type : "png",
    pixelRatio : 2, //放大 2 倍
    backgroundColor : '#fff'
});
img.onload = function() {
    var canvas =
document.createElement("canvas");
    canvas.width = img.width;
    canvas.height = img.height;
    var ctx = canvas.getContext('2d');
    ctx.drawImage(img, 0, 0);
    var dataURL =
canvas.toDataURL('image/png');
    var a = document.createElement('a');
    // 创建一个单击事件
```

```
var event = new MouseEvent('click');
```

// 将 a 的 download 属性设置为我们想要下载的图片名称，若 name 不存在则使用‘下载图片名称’作为默认名称

```
a.download = 'storeage_manage.png';  
// 将生成的 URL 设置为 a.href 属性  
a.href = dataURL;  
// 触发 a 的单击事件  
a.dispatchEvent(event);  
};  
}
```

```
function expXls() {  
$('#datatable').table2excel({  
    exclude : '.noExl',  
    name : 'Excel Document Name',  
    filename : 'storeage_t_data.xls',  
    exclude_img : true,  
    exclude_links : true,  
    exclude_inputs : true  
})  
}
```

```
function doPrint() {
```

```
window.print();  
  
}  
  
$(function() {  
findGrid()  
  
})
```

其他功能就不贴了，在同目录。

2) 发货管理

```
@RequestMapping(params = "deliver", method =  
{RequestMethod.GET, RequestMethod.POST})  
  
    public void deliver(HttpServletRequest request,  
        HttpServletResponse response, ModelMap model)  
        throws Exception {  
        try {  
            VelocityContext velocityContext = new  
VelocityContext();  
  
            System.out.println("deliver");  
  
            String viewName = "fresh/main/deliver.vm";  
  
            ViewVelocity.view(request, response,  
viewName, velocityContext);  
  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



```
}  
  
}
```

数据查询代码:

```
@Controller  
  
@RequestMapping("/fresh/fDOrder")  
  
public class FDOrderController  
  
.....  
  
/**  
 * Find store grid data  
 *  
 * @return  
 */  
  
@RequestMapping(params = "findGrid", method =  
{RequestMethod.GET, RequestMethod.POST})  
  
@ResponseBody  
  
public AjaxJson findGrid(@ModelAttribute DeliverVo  
deliverVo) {  
  
    AjaxJson j = new AjaxJson();  
  
    LOG.info("{} ", deliverVo);  
  
    try {  
  
        List<OrderCountResult> list =  
        fDQueryDao.getOrderCountWithDate(deliverVo);
```

```

        j.setObj(list);
        j.setMsg("成功");
    } catch (Exception e) {
        j.setSuccess(false);
        j.setMsg("失败");
        e.printStackTrace();
    }
    return j;
}

```

View:

```

/jeecg-p3-starter-web/src/main/resources/content/fresh/main/deliver.vm

```

3) 运配管理

```

@RequestMapping(params = "distribution", method
= {RequestMethod.GET, RequestMethod.POST})
    public void distribution(HttpServletRequest request, HttpServletResponse response, ModelMap model)
        throws Exception {
        try {

```

```

        VelocityContext velocityContext = new
VelocityContext();

        System.out.println("distribution");

        String viewName =
"fresh/main/distribution.vm";

        ViewVelocity.view(request, response,
viewName, velocityContext);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

数据查询代码：

```

@Controller
@RequestMapping("/fresh/fDOrder")
public class FDOrderController
.....

/**
 * Find store grid data
 *
 * @return
 */

@RequestMapping(params = "findBad", method =

```

```

{RequestMethod.GET, RequestMethod.POST})

@ResponseBody

public AjaxJson findBad(@ModelAttribute DeliverVo
deliverVo) {

    AjaxJson j = new AjaxJson();

    LOG.info("{} ", deliverVo);

    try {

        List<OrderBadTemResult> list =
fdQueryDao.getBadOrder(deliverVo);

        LinkedHashMap<Double, BadInfo> map = new
LinkedHashMap<>();

        for (OrderBadTemResult or : list) {

            Double t = or.getTranstemperature();

            Integer gobad = or.getGobad();

            if (!map.containsKey(t)) {

                map.put(t, new BadInfo());

            }

            BadInfo info = map.get(t);

            if (gobad == 0) {

                info.addGoodcount();

            } else {

                info.addBadcount();

```

```
    } ;  
}  
ArrayList<Double> xdata = new ArrayList<>();  
ArrayList<Double> sdata = new ArrayList<>();  
ArrayList<JSONObject> tdata = new  
ArrayList<>();  
map.forEach((k, v) -> {  
    xdata.add(k);  
    sdata.add(v.getRate());  
    JSONObject js = new JSONObject();  
    js.put("temp", k);  
    js.put("rate", v.getRate());  
    tdata.add(js);  
});  
JSONObject res = new JSONObject();  
res.put("xdata", xdata);  
res.put("sdata", sdata);  
res.put("tdata", tdata);  
j.setObj(res);  
j.setMsg("成功");  
} catch (Exception e) {  
    j.setSuccess(false);
```

```

        j.setMsg("失败");
        e.printStackTrace();
    }

    return j;
}

```

View:

/jeecg-p3-starter-web/src/main/resources/content/fresh/main/distribution.vm

4) 退货管理

```

@RequestMapping(params = "backgoods", method =
{RequestMethod.GET, RequestMethod.POST})

    public void backgoods(HttpServletRequest request, HttpServletResponse response, ModelMap model)

        throws Exception {

        try {

            VelocityContext velocityContext = new
VelocityContext();

            System.out.println("backgoods");

            String viewName =
"fresh/main/backgoods.vm";

```

```

        ViewVelocity.view(request, response,
viewName, velocityContext);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

数据查询代码:

```

@Controller
@RequestMapping("/fresh/fDOrder")
public class FDOrderController
.....

/**
 * Find store grid data
 *
 * @return
 */
@RequestMapping(params = "findBack", method =
{RequestMethod.GET, RequestMethod.POST})
@ResponseBody
public AjaxJson findBack(@ModelAttribute DeliverVo
deliverVo) {

```

```
    AjaxJson j = new AjaxJson();  
    LOG.info("{} ", deliverVo);  
    try {  
        List<OrderBackResult> list =  
fdQueryDao.getBack(deliverVo);  
        j.setObj(list);  
        j.setMsg("成功");  
    } catch (Exception e) {  
        j.setSuccess(false);  
        j.setMsg("失败");  
        e.printStackTrace();  
    }  
    return j;  
}
```

View:

```
/jeecg-p3-starter-web/src/main/resources/content/f  
resh/main/backgoods.vm
```


三、 技术架构图

