目录：

数据操作；

页面；

·

·分页；

//////////////////////////////////////////////////////////////////

数据操作；

数据绑定机制：

JSF提供了数值绑定机制，使得在Faces标记中访问bean property 很容易，其原理是用 JSF-EL （表达式语言）把 JSF 组件绑定到受管理的 bean 属性。

·action中声明变量；

·页面，绑定action的变量；

参数传递：

普通表单的数据添加、修改，都是通过实体类来传递页面数据；

~一些特殊的，通过action中声明参数，页面隐藏控件来传值；

主页面，可以直接取到include页面的值，使用跟include一样的参数；

1，参数传递：

Action中，声明变量，在页面和action间传递值；

**private** String borrowId;//借款记录id

**private** String resultMsg;//返回值

get/set下；

~

Id，直接使用；

页面加载时，控件赋值：

~从后台绑定时，给出绑定参数(声明变量，实体类，可以有多个实体类，然后赋值)

页面，

<a4j:commandButton id=*"save"* value=*"保存"*

action=*"#{borrowMoneyAction.save}"*

execute=*"@form"*

data=*"#{borrowMoneyAction.resultMsg}"*

onclick="return validApprove();"

oncomplete=*"parent.waitHide();afterSaveDo2(event.data,#{rich:component('approveBorrowMoney')});"*>

<a4j:param value=*"getCheckboxIds('borrowSearchForm:borrowId')"* assignTo=*"#{borrowMoneyAction.borrowId}"* noEscape=*"true"* />

</a4j:commandButton>

项目业务逻辑类，action：---普通实现，

//日志加入---引入需要的service---需要的实体类---声明需要的变量---service、实体类、变量等都需要get/set---

@ManagedBean(name = "bankCardLockSelectAction")

@ViewScoped //一般都是视图范围bean

**public** **class** BankCardLockSelectAction **implements** Serializable {

**private** **final** **static** Logger ***logger*** = Logger.*getLogger*(BankCardLockSelectAction.**class**);

@ManagedProperty(value = "#{userBindBankServiceImpl}")

UserBindBankService userBindBankService;

BankInfoVo bankInfoVo = **new** BankInfoVo();

/\*\* 提示信息 \*/

**private** String resultMsg; //需要get/set

----------------------------

页面加载时查询并绑定内容：

//execute=*"@form"，带参数；不带参数的使用*execute=*"@this"*

<a4j:commandButton id=*"btnSearchFirstAccount"* value=*"当前直通车开通金额的总额"*

onclick=""

action=*"#{firstTenderRealAction.getCount()}"*

execute=*"@form"*

data=*""*

oncomplete=*""*

render=*"firstTenderRealAccount"* style="display:*none*">

</a4j:commandButton>

当前直通车开通金额的总额为：

<span style="color:*red*">

<h:outputText id=*"firstTenderRealAccount"* value=*"#{firstTenderRealAction.firstTenderRealAccount}"* >

<f:convertNumber currencySymbol=*"$"* type=*"number"*/>

</h:outputText>

</span>

元

Js：

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

//查询开通金额

$('[id="firstTenderRealForm:btnSearchFirstAccount"]').trigger("click");

});

Action：

//声明变量，get/set；

//直通车开通金额的总额

**private** BigDecimal firstTenderRealAccount;

**public** **void** getCount(){

//直通车开通金额的总额计算

firstTenderRealAccount = firstTenderRealService.queryFirstTenderRealAccount(firstTenderRealCnd);

}

然后的是：

数据操作的查询方法；

////////////////////////////////////////////////////////////////////

页面：

主内容：

框架页面：没有框架页面，只是每个页面使用ui:include包含需要的页面；

分页；

消息提示，---返回普通提示；返回异常信息；返回自定义异常提示信息；

权限；

~

页面-控件用法；

后台框架页面，islogin.xhtml

都是通过<ui:include src="dailyofficeworknav.xhtml"></ui:include>加入其它页面；

顶部，top.xhtml；

左侧菜单，*main.xhtml*

右侧，iframe；

/page/userbindbank/bankcard\_lock\_list.xhtml:

登录：

登录页面，

A4j按钮；输入验证；返回登录错误提示；

~

loginAction，

声明变量：用户名，密码，错误提示信息；

//shiro认证，

**public** String login(){

String username = **this**.getUsername();

String password = **this**.getPassword();

UsernamePasswordToken token = **new** UsernamePasswordToken(username, password);

Subject currentUser = SecurityUtils.*getSubject*();

**try** {

currentUser.login(token);

//成功，跳转页面；

} **catch** (UnknownAccountException e) {

resultMsg = "用户名不存在";

}

//失败，返回错误提示；

return null;

}

//登出 | shiro方式；

**public** String logout(){

Subject currentUser = SecurityUtils.*getSubject*();

**try** {

currentUser.logout();

**return** "/index.xhtml?faces-redirect=true";

} **catch** (Exception e) {

log.error(e.getMessage());

}**finally**{

**return** "/index.xhtml?faces-redirect=true";

}

}

action权限检测：

页面url,action---配置文件(web.xml, faces-config.xml), ActionFilter类；

类中授权；

-------------------------------

分页：

声明一个page对象，保存分的页码等相关传递参数；

页面四个按钮，action事件加载数据，携带参数；

页面：

<div style="margin-top:*5px*;">

共 &nbsp; #{bankCardLockSelectAction.page.totalPage} &nbsp; 页 &nbsp;共 &nbsp; #{bankCardLockSelectAction.page.totalCount} &nbsp; 条

第 &nbsp; #{bankCardLockSelectAction.page.pageNo} / #{bankCardLockSelectAction.page.totalPage} &nbsp; 页

<a4j:commandButton id=*"firstpage"* name=*"lastpage"* value=*"首页"*

action=*"#{bankCardLockSelectAction.searchBankCardLockList}"* execute=*"@this"*

data=*"#{bankCardLockSelectAction.page}"* render=*"bankCardLockList"*>

<a4j:param name=*"pageNo"* value=*"0"*

assignTo=*"#{bankCardLockSelectAction.pageNo}"* />

</a4j:commandButton>

&nbsp;

<a4j:commandButton id=*"prevpage"* name=*"prevpage"* value=*"上一页"*

action=*"#{bankCardLockSelectAction.searchBankCardLockList}"* execute=*"@this"*

data=*"#{bankCardLockSelectAction.page}"* render=*"bankCardLockList"*>

<a4j:param name=*"pageNo"*

value=*"#{bankCardLockSelectAction.page.pageNo-1}"*

assignTo=*"#{bankCardLockSelectAction.pageNo}"* />

</a4j:commandButton>

&nbsp;

<a4j:commandButton id=*"nextpage"* name=*"nextpage"* value=*"下一页"*

action=*"#{bankCardLockSelectAction.searchBankCardLockList}"* execute=*"@this"*

data=*"#{bankCardLockSelectAction.page}"* render=*"bankCardLockList"*>

<a4j:param name=*"pageNo"*

value=*"#{bankCardLockSelectAction.page.getNextPage()}"*

assignTo=*"#{bankCardLockSelectAction.pageNo}"* />

</a4j:commandButton>

&nbsp;

<a4j:commandButton id=*"lastpage"* name=*"lastpage"* value=*"末页"*

action=*"#{bankCardLockSelectAction.searchBankCardLockList}"* execute=*"@this"*

data=*"#{bankCardLockSelectAction.page}"* render=*"bankCardLockList"*>

<a4j:param name=*"pageNo"*

value=*"#{bankCardLockSelectAction.page.totalPage}"*

assignTo=*"#{bankCardLockSelectAction.pageNo}"* />

</a4j:commandButton>

</div>

Action：

Page page = **new** Page();

**private** **final** **int** pageSize = 10;

**private** **int** pageNo;

//查询参数实体类；

BankCardLockCnd bankCardLockCnd = **new** BankCardLockCnd();

**public** **void** searchBankCardLockList() {

**try** {

page = userBindBankService.queryBankCardLockListByCnd(bankCardLockCnd, pageNo, pageSize);

serviceImpl：

**public** Page queryBankCardLockListByCnd(BankCardLockCnd bankCardLockCnd, **int** pageNo, **int** pageSize) {

Page page = **new** Page(pageNo, pageSize);

page.setResult(bankinfoLogMapper.queryPageAllBankCardLockListByCnd(bankCardLockCnd, page));

page.setTotalCount(bankinfoLogMapper.queryCountPageAllBankCardLockListByCnd(bankCardLockCnd));

**return** page;

}

----------------------------------

消息提示，---返回普通提示；返回异常信息；返回自定义异常提示信息；

页面，---接收提示参数；

<a4j:commandButton id=*"saveSubmit"* value=*"提交"*

action=*"#{financialContractAction.save}"* execute=*"@form"*

data=*"#{financialContractAction.resultMsg}"*

onclick="setContractStatus1(7);return valid3();"

oncomplete=*"afterSave1(event.data);"*>

</a4j:commandButton>

action中，

1. ，普通提示信息，

resultMsg = "合同编号已存在,请重新填写. ";

2），

数据操作在try-catch中，

**try** {

financialContractService.save(financialContract);

} **catch** (Exception e) {

e.printStackTrace();

log.error(e.getMessage());

resultMsg = "保存失败，请核对填写的数据";

**return**;

}

3），更详细的自定义提示信息，并整合异常信息，

Map<String,Object> resultMap=new HashMap<String,Object>();

**try** {

//数据操作的结果，返回提示； approveNewBorrowService.saveNewBorrow(Long.*parseLong*(borrowId),okflag,memo,resultMap);

resultMsg=String.*valueOf*(resultMap.get("msg"));

} **catch** (Exception e) {

resultMsg=e.getMessage(); //可以接收到下面service中自定义的异常类信息；

//可以自己写异常；

//e.printStackTrace();

//resultMsg = "保存失败，请核对填写的数据";

log.error(e.getMessage());

}

Service中，自己写异常类：

**public** **void** saveBorrowMoney(**long** borrowId, **int** okflag, String memo,

Map<String, Object> resultMap) **throws** FuiouModException, Exception {

//成功，返回提示；

resultMap.put("msg", "放款成功");

} **else** {

**throw** **new** Exception("数据已过期,请刷新后重试");

}

自定义异常类，

**public** **class** FuiouModException **extends** Exception{

/\*\*

\* **@fields** serialVersionUID

\*/

**private** **static** **final** **long** ***serialVersionUID*** = -6597013124481645689L;

**private** String msg;

**public** FuiouModException(String msg) {

**this**.msg=msg;

}

/\* (non-Javadoc)

\* @see java.lang.Throwable#getMessage()

\*/

@Override

**public** String getMessage() {

**return** **this**.msg;

}

}

/////////////////////////////////////////////////////////////////////////

页面项目实际：

//一个面板，里面一个表格，表格循环数据；| <a4j:repeat

<rich:panel>

<h:panelGroup id=*"bankCardLockList"*>

<table class=*"fulltable"* style="width:*99%*;">

<tr class=*"tr\_head"*>

<th width=*"5%"*>序号</th>

<th width=*"10%"*>计划代收日期</th>

<th width=*"10%"*>代收本金</th>

<th width=*"10%"*>代收利息</th>

<th width=*"10%"*>代收总额</th>

<th width=*"10%"*>代收状态</th>

<th width=*"10%"*>复核结果</th>

<th width=*"31%"*>复核人</th>

</tr>

<a4j:repeat rowKeyVar=*"i"* value=*"#{borrowAction.collectionList}"* var=*"item"*>

<tr class=*'tr\_#{i%2}'*>

<td>#{i+1}</td>

<td><h:outputText value=*"#{item.planDate}"*><f:convertDateTime pattern=*"yyyy-MM-dd HH:mm:ss"* /></h:outputText></td>

<td>#{item.planPrinciple}</td>

<td>#{item.planInterest}</td>

<td>#{item.planTotal}</td>

<td>#{item.status}</td>

<td>#{item.approveResult}</td>

<td>#{item.approveName}</td>

</tr>

</a4j:repeat>

</table>

<div style="height:*20px*;"></div>

</rich:panel>

文本控件+隐藏控件：

文本控件显示不可修改值，隐藏控件传递值到action；

<h:outputText id=*"shownewCalculationDate"*

value=*"#{contractApplyAction.financialExamination.calculationDate}"*>

<f:convertDateTime pattern=*"yyyy-MM-dd"*></f:convertDateTime>

</h:outputText>

<h:inputText id=*"newCalculationDate"*

style="display:*none*"

value=*"#{contractApplyAction.financialExamination.calculationDate}"*>

<f:convertDateTime pattern=*"yyyy-MM-dd"*></f:convertDateTime>

</h:inputText>

~

另，使用<h:inputHidden

<h:outputText id=*"shownewenddate"*

value=*"#{contractApplyAction.financialExamination.redemptionDate}"*>

<f:convertDateTime pattern=*"yyyy-MM-dd"*></f:convertDateTime>

</h:outputText>

<h:inputHidden id=*"newenddate"*

value=*"#{contractApplyAction.financialExamination.redemptionDate}"* >

<f:convertDateTime pattern=*"yyyy-MM-dd"*></f:convertDateTime>

</h:inputHidden>

////////////////////////////////////////////////////////////////////

输入验证：---手写jquery； | 另，考虑jquery插件使用；---和jquery插件的验证本质同；

1， 页面控件，按钮：

<h:head>

<script src=*"${path}/js/jquery1.42.min.js"* type=*"text/javascript"*></script>

</h:head>

<h:form id=*"contractAddForm"*>

<h:inputText id=*"contractCode"* class=*"realInput"* value=*"卡号"* maxlength=*"50"* />

<a4j:commandButton id=*"save"* value=*"保存"* onclick="return valid1();">

</a4j:commandButton>

</h:form>

2，<h:outputScript>……</h:outputScript>中写js代码；

3，Js代码，

function valid1(){

var reg; //定义变量，进行比较

var value;

var regs=/^(\d{0,18})$/ //正则表达式变量

reg=/\S/;

value=$('[id="contractAddForm:contractCode"]').val(); //获得变量

if(!reg.test(value)){

alert("请输入合同编号");

return false;

}

if(!regs.test(value)){

alert("请输入有效的合同编号,只能为数字");

return false;

}

if(value.length>20){

alert("合同编号过长");

return false;

}

//验证证件号码

var idNumber=$('[id="contractAddForm:idNumber"]').val();//证件号码

reg = /(^\d{15}$)|(^\d{18}$)|(^\d{17}(\d|X|x)$)/;

if(!reg.test(idNumber)){

alert("身份证号不合法");

return false;

}

var phoneNumber = $('[id="contractAddForm:phoneNumber"]').val();//移动电话

//验证移动电话

reg=/^\d{11}$/;

if(!reg.test(phoneNumber)){

alert("请输入11位手机号码");

return;

}

//验证邮箱

reg=/\S/;

value=$('[id="contractAddForm:email"]').val();//电子邮箱

if(reg.test(value)){

reg= /^([a-zA-Z0-9\_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

if(!reg.test(value)){

alert("请输入合法邮箱");

return;

}

}

//年化利率

reg=/(^[0-9]$|^[1][0-9]$|20)|(^([0-9]|[1][0-9]|20)[\.][0-9]{1,8}$)/;

value = $('[id="contractAddForm:contractRate"]').val();

if(!reg.test(value)){

alert("请输入正确的年化利率（0-20最多8位小数）");

return false;

}

//出借资金(人民币)

//reg=/^\d{0,20}$/;

reg=/^([1-9][\d]{0,7}|0)(\.[\d]{1,2})?$/;

value=$('[id="contractAddForm:contractFunds"]').val();

if(!reg.test(value)){

alert("请输入合法 出借资金");

return false;

}

if(value &lt;50000) {

alert("出借资金不能低于50000");

return false;

}

if(value &gt;50000000) {

alert("出借资金不能大于五千万");

return false;

}

}

--------------------------------------------------------

列表选择按钮：

<h:inputHidden id="collectionId" /> //隐藏控件，

<a4j:commandButton value=*"查看详情"* styleClass=*"longbutton"* execute=*"@this"* render=*"detailBorrowCollectionForm"*

onclick="return beforeSingleOperation('collectionSearchForm:collectionId');"

action=*"#{borrowCollectionAction.detail}"* oncomplete=*"showPanel(#{rich:component('detailBorrowCollectionPopup')});"* >

<a4j:param value=*"getCheckboxIds('collectionSearchForm:collectionId')"* assignTo=*"#{borrowCollectionAction.collectionId}"* noEscape=*"true"* />

</a4j:commandButton>

//列表数据，input-checkbox控件，

<a4j:repeat rowKeyVar=*"i"* value=*"#{borrowCollectionAction.pages.list}"* var=*"bc"*>

<tr class=*'tr\_#{i%2}'*>

<td width=*"1%"*><input type=*"checkbox"* name=*"checkbox"* value=*"#{bc.id}"*/></td>

<td>#{i+1}</td>

<td>#{bc.borrowNo}</td>

//js，按钮选择，

function beforeSingleOperation (idName){

var id\_array=new Array();

var i=0;

$('input[name="checkbox"]:checked').each(function(){

id\_array.push($(this).val());//向数组中添加元素

i++;

});

if(i==0){

alert("请选择一条记录。");

return false;

}else if(i>1){

alert("只能选择一条记录。");

return false;

}

var idstr=id\_array.join(',');//将数组元素连接起来以构建一个字符串

//赋值到页面隐藏控件

$('[id="'+idName+'"]').val(idstr);

return true;

}

//获得隐藏控件的值

**function** getCheckboxIds(idName){

**return** $('[id="'+idName+'"]').val();

}

//删除

<a4j:commandButton value=*"删除"* styleClass=*"button"*

oncomplete=*"afterDel();"* onclick="return beforeBatchDelete ('collectionSearchForm:collectionId');"

function beforeBatchDelete (idName){

var id\_array=new Array();

var i=0;

$('input[name="checkbox"]:checked').each(function(){

id\_array.push($(this).val());//向数组中添加元素

i++;

});

if(i==0){

alert("请至少选择一条记录。");

return false;

}

var idstr=id\_array.join(',');//将数组元素连接起来以构建一个字符串

$('[id="'+idName+'"]').val(idstr);

if (confirm("确定要删除?")) {

return true;

}

return false;

}

--------------------------------------------

弹出层或跳转页面：

//列表.xhtml---详情按钮，

<a4j:commandButton value=*"查看详情"* styleClass=*"longbutton"* execute=*"@this"* render=*"detailBorrowCollectionForm"*

onclick="return beforeSingleOperation('collectionSearchForm:collectionId');"

action=*"#{borrowCollectionAction.detail}"* oncomplete=*"showPanel(#{rich:component('detailBorrowCollectionPopup')});"* >

<a4j:param value=*"getCheckboxIds('collectionSearchForm:collectionId')"* assignTo=*"#{borrowCollectionAction.collectionId}"* noEscape=*"true"* />

</a4j:commandButton>

//Js，

**function** showPanel(panel){

panel.show();

}

---通过h:form的id找到页面，

//详情.xhtml

<h:form id=*"detailBorrowCollectionForm"* >

<rich:popupPanel id=*"detailBorrowCollectionPopup"* height=*"450"* minWidth=*"1050"* left=*"20px;"* top=*"1"* style="width:*100%*;height:*95%*;padding:*0px*;"

modal=*"true"* resizeable=*"true"* domElementAttachment=*"form"*>

<f:facet name=*"header"*>

<font><h:outputText id=*"edittabletitle"* value=*"代收详情"* class=*"listzen"* /></font>

</f:facet>

<f:facet name=*"controls"*>

<font><h:commandLink value=*"关闭"* class=*"closeMoreInfo"* onclick="#{rich:component('detailBorrowCollectionPopup')}.hide();return false;"/></font>

</f:facet>

<ui:include src=*"detailBorrowCollectionDiv.xhtml"*></ui:include>

</rich:popupPanel>

</h:form>

跳转页面：

页面，

<a4j:commandButton value=*"新增"* styleClass=*"button"*

action=*"#{financialContractAction.add}"* execute=*"@this"* data=*"#{financialContractAction.forUrl}"* oncomplete=*"doAddPopup(event.data);"* render=*"contractAddForm"*></a4j:commandButton>

action中，

**private** String forUrl;

**this**.setForUrl("/page/oa/contract/add.xhtml?faces-redirect=true");

function doAddPopup(urlstr){

parent.changesrc(urlstr);

//location.href=urlstr;

}

*----------------------*

*日期计算：*

*合同签订日期，利息计算日期，到期支付日期---合同签订日期输入后自动生成后两者；*

*日期公共类，引用类文件，使用方法；---见“js,jquery日期函数，公共类等.docx”；*

*页面，---onchange()事件，*

<h:inputText styleClass=*"Wdate"* id=*"newstartdate"* onclick="WdatePicker()"

value=*"#{contractApplyAction.financialExamination.signatureDate}"*

onchange=*"return changeredemptionDate(#{contractApplyAction.financialExamination.isMonthly});"*>

<f:convertDateTime pattern=*"yyyy-MM-dd"*></f:convertDateTime>

</h:inputText>

**function** changeredemptionDate(ismonthly){

//页面取值，

**var** oldstart=$('[id="examinationstartenddetailForm:calculationDate"]').text(); **var** oldend=$('[id="examinationstartenddetailForm:redemptionDate"]').text();

**var** thisstart=$('[id="examinationstartenddetailForm:newstartdate"]').val();

//加一天

**var** calculationDate=dateToStr(addDate(strToDate(thisstart),1));

$('[id="examinationstartenddetailForm:newCalculationDate"]').val(calculationDate);

//签订日期为空时，置空其它；

**if**(thisstart==""){

$('[id="examinationstartenddetailForm:newenddate"]').val("");

$('[id="examinationstartenddetailForm:newCalculationDate"]').val("");

**return**;

}

//计算到期支付日期；

**if**(ismonthly==2){

$('[id="examinationstartenddetailForm:newenddate"]').val(dateToStr(addDate(strToDate(thisstart),daysNum(strToDate(oldend),strToDate(oldstart)))));

}**else**{

$('[id="examinationstartenddetailForm:newenddate"]').val(dateToStr(addMonths(strToDate(thisstart),monthsNum(strToDate(oldend),strToDate(oldstart)))));

}

}

利息计算：---自动计算利息，本金合计；

--------------------

页面选择，传递到另一页面；

//页面按钮，

<a4j:commandButton value=*"导入"*

style="width:*50px*;" onclick="loadBorrow2();" />

//js，---弹出窗体-页面，

<h:outputScript>

function loadBorrow2(){

#{rich:component('loadBorrowPopup')}.show();

}

//可导入合同列表页面中，loadContract.xhtml

<input type=*"checkbox"* name=*"checkbox2"* value=*"#{contract.id}"* />

<a4j:commandButton value=*"确认导入"* styleClass=*"button"*

execute=*"@form"*

action=*"#{creditorAction.importcontract}"*

onclick="return beforeimport();"

oncomplete=*"afterimport()"*

render=*"creditorapplyForm:creditorinfo"*

>

<a4j:param value=*"getSelectContractId()"*

assignTo=*"#{creditorAction.financialContract.id}"*

noEscape=*"true"* />

</a4j:commandButton>

·按钮操作，action，

//js，

function beforeimport(){

var id\_array=new Array();

var i=0;

$('input[name="checkbox2"]:checked').each(function(){

id\_array.push($(this).val());//向数组中添加元素

i++;

});

if(i==0){

alert("请选择一条记录");

return false;

}else if(i>1){

alert("只能选择一条记录");

return false;

}

var idstr=id\_array.join(',');//将数组元素连接起来以构建一个字符串

$('[id="loadBorrowForm:selectContractId"]').val(idstr);

}

function getSelectContractId(){

return $('[id="loadBorrowForm:selectContractId"]').val();

}

function afterimport(){

#{rich:component("loadBorrowPopup")}.hide();

}

//////////////////////////////////////////////////////