目录：

数据操作；

页面；

输入验证；

日期计算；

利息计算；

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

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：

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；

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；

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

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;

}

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();

}

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

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;

}

}

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

*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；

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