重复提交处理：

前台；

后台；

Struts2的令牌机制；或Springmvc token机制；

适用：

所有表单提交的操作，都需要重复提交处理；前台js，后台代码(包括，可能在存储过程中处理)；

//让当前的线程睡眠4秒钟，模拟网络延迟而导致表单重复提交的现象

Thread.*sleep*(4\*1000);

总：后台2种：

1，session令牌； ---应用于，单纯添加； | 优先使用；

2，insert ...where not exists； ---少用；

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

前台：

1，js中处理，点击按钮后禁用按钮，防止再次点击；

1），按钮：

<input onclick="goto\_register()" type=*"button"* name=*"Submit1"* id=*"subbtn"* class=*"login\_btn"* value=*"提交注册"* />

$('#subbtn').attr("disabled","disabled");

恢复，

$('#subbtn').removeAttr("disabled");

或，

$("#btnVIPAppro").removeAttr("onclick");

$("#btnVIPAppro").attr("onclick","forsubmit();"); //恢复；

2），超链的：

<a id=*"registerHref"* href=*"javascript:void(0);"* onclick="goto\_register();">提交注册</a>

$('#registerHref').attr("onclick", "");

恢复，

$('#registerHref').attr("onclick", "goto\_register();");

1. ，加进度条的方式也有效果，

**function** saveCommentCon() {

**var** \_load = layer.load('处理中..');

success : **function**(data) {

...

layer.close(\_load);

后台：

方式1：session令牌方式；

·进入页面前，| 创建session，页面控件绑定值；

String token = System.*currentTimeMillis*()+**new** Random().nextInt()+"";

request.getSession().setAttribute("token", token);

mav.addObject("c\_token", token);

·页面，

<input type=*"hidden"* id=*"c\_token"* name=*"c\_token"* value=*"*${c\_token}*"*>

·Js，传递参数，

$.ajax({

url : "${path}/bankinfo/lockBankCard.do",

data : {'c\_token' : c\_token

·后台添加操作，

//锁定银行卡新增功能 | 比较页面传递的参数和session中的值，

**public** MessageBox lockBankCard(HttpServletRequest request,HttpSession session) {

String c\_token = request.getParameter("c\_token");

String s\_token = (String) request.getSession().getAttribute("token");

**try** {

**if** (c\_token!=**null** && s\_token!=**null**) {

**if**(c\_token.equals(s\_token)){

msg = bankInfoService.insertBankCardLock(shiroUser.getUserId());

**if** (msg.equals("success")) {

**return** MessageBox.*build*("1", "锁定成功.");

} **else** {

**return** MessageBox.*build*("0", msg);

}

}

**else** { //确实是重复提交导致的两者不等，

**return** MessageBox.*build*("0", "不能重复提交");

}

}**else**{

**return** MessageBox.*build*("0", "锁定失败..");

}

} **catch** (Exception e) {

logger.error("锁定银行卡失败.", e);

msg = "锁定失败";

e.printStackTrace();

}**finally**{

request.getSession().removeAttribute("token");

}

方式2，insert ...where not exists：

//存在userid相同，已有记录type=0,status=0，则不能再添加；

insert into rocky\_bankinfo\_log(USER\_ID, CARDNUM, TYPE, STATUS, ADD\_BY, ADD\_TIME, REMARK)

select \* from (

select 20884 as USER\_ID, '0' as CARDNUM, 0 as TYPE, 0 as STATUS, 0 as ADD\_BY,'2014-12-2','系统锁定'

) t

where not exists (select 1 from rocky\_bankinfo\_log c where c.USER\_ID = t.USER\_ID and c.TYPE = 0 and c.STATUS = 0);

~

<insert id=*"insert"* parameterType=*"com.cxdai.base.entity.BankinfoLog"* >

insert into rocky\_bankinfo\_log (USER\_ID, CARDNUM,

TYPE, STATUS, ADD\_BY, ADD\_TIME,

REMARK)

select \* from (

select #{userId,jdbcType=INTEGER} as USER\_ID, #{cardNum,jdbcType=VARCHAR} as CARDNUM,

#{type,jdbcType=TINYINT} as TYPE, #{status,jdbcType=TINYINT} as STATUS, #{addBy,jdbcType=INTEGER} as ADD\_BY,

#{addTime,jdbcType=TIMESTAMP} as ADD\_TIME,#{remark,jdbcType=VARCHAR} as REMARK

) t

where not exists (select 1 from rocky\_bankinfo\_log c where c.USER\_ID = t.USER\_ID and c.TYPE = 0 and c.STATUS = 0);

</insert>

~

实例：单机重复提交可以控制。 双机，各开一个浏览器，同时提交，不能控制。

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