重复提交处理：

前台；

后台；

Struts2的令牌机制；

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)；

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

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();");

后台：

只有有并发冲突的，才需要如此操作，没有的不需要；

1，发起债权转让的投标记录，不能重复发起； | 本质是修改添加，使用加锁方式；

方式1：sql语句加锁；

查询数据，加锁；

返回查询结果，不为空，操作该语句；---完成自动解锁语句。

<select id=*"lockCollectionRecordByTenderId"* parameterType=*"java.lang.Integer"* resultMap=*"com.cxdai.portal.invest.mapper.AccCommonCollectionMapper.accCommonCollectionResultMap"* >

SELECT \* FROM rocky\_b\_collectionrecord rbc WHERE rbc.TENDER\_ID = #{tenderId} and rbc.`ORDER`=#{transferBeginOrder} and rbc.STATUS=0 LIMIT 0,1 for update

</select>

// 锁住代收表中 第一条待还记录 的期数是不是原先期数 ，数据库加锁防止并发 转让操作，事务不结束，锁不解开

CommonCollectionVo conCollectionVo = lockCollectionRecordByTenderId(bTransferVo.getTenderId(), bTransferVo.getTransferBeginOrder().intValue());

// 判断期数 和 转让期数是不是同一期

**if** (conCollectionVo == **null**) {

**throw** **new** RuntimeException("该笔代收已经不存在,不能发起转让");

}

**if** (existsTransferRecordBytenderId(bTransferVo.getTenderId())) {

**throw** **new** RuntimeException("该投标已经被转让");

}

// 插入转让记录

bTransferMapper.insert(bTransferVo);

**private** CommonCollectionVo lockCollectionRecordByTenderId(Integer tenderId, Integer transferBeginOrder) {

**return** bTransferMapper.lockCollectionRecordByTenderId(tenderId, transferBeginOrder);

}

方式2：悲观锁；代码中加锁；

2，单纯添加的操作：

方式1： //存在userid相同，已有记录type=0,status=0，则不能再添加；

//代码中的先查询是否存在数据(查询userid,type,status)，没控制住；

mybatis的xml中，添加语句执行前，先判断是否有锁定语句，可以保证去除重复；

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>

方式2：session令牌方式；

实例：

债权转让撤销；---防止重复；

实现在存储过程中；

可重复判断：存储过程中，执行前，判断状态，已撤销的，跳出；---就不会重复了。