Atitit mybatis缓存遇到的问题与解决

[1.1. 只有getobj 没有执行putobject 1](#_Toc27113)

[1.2. 默认所有的select要读取缓存怎么办。。执行过滤 可以自己定义个白名单。 2](#_Toc29659)

[1.3. 默认namespace up del操作清空缓存，做了拦截不让他自动清空，不让缓存失效。使用时间到期清空模式 3](#_Toc30386)

[2. 缓存内容与执行顺序不对 增加synchronized 锁定 3](#_Toc22102)

[2.1. 可能没有走缓存，导致curkey还是上次的。每次put玩clr curkey 并put增加sync锁，保证走完 3](#_Toc24671)

[2.2. 每次getxx执行前去除curkey thredlocal遍量 3](#_Toc25370)

[2.3. 增加keytag 教研校验 这个额 keytag 是mybati配置文件里面的 nanespace.sqlid 4](#_Toc25940)

## 只有getobj 没有执行putobject

估计是因为没有提交，所以还没有putobj ，应该会自己putobj的，如果不能，那么需要自己在业务里面增加

可能是spring了但是却取消了事务模式。。所以没有提交

//ati set cache

**new** MybatisRedisCache().putObjectByThrdlocalCurKey( uMap);

//end set cache finish

**public** **void** putObjectByThrdlocalCurKey(Object uMap) {

Object **curkey**= MybatisRedisCache.*curkey*.get();

putObject(curkey,uMap);

}

**public** **static** ThreadLocal<Object> *curkey*;

// 获取员工编号和姓名

Map **uMap** = getOperation(map);

//ati set cache

Object **curkey**= MybatisRedisCache.*curkey*.get();

**new** MybatisRedisCache().putObject(curkey, uMap);

## 默认所有的select要读取缓存怎么办。。执行过滤 可以自己定义个白名单。

白名单机制，不然每个select都要默认走，，要一个个加麻烦。。Usecache=false麻烦 还有黑名单机制

@Override

**public** **void** putObject(Object key, Object value) {

// Object curkey= MybatisRedisCache.curkey.get();

String **string\_curkey** = key.toString().toLowerCase();

**if**(string\_curkey.contains("tmp") || string\_curkey.contains("temp") || string\_curkey.contains("count") || string\_curkey.contains("insert") || string\_curkey.contains("update") || string\_curkey.contains("delete") || string\_curkey.contains("drop") || string\_curkey.contains("exist"))

**return**;

**try** {

*logger*.debug(">>>>>>>>>>>>>>>>>>>>>>>>putObject key:" + key + " ,val:" + value);

**if**(key==**null**)**return**;

Jedis **Jedis1**= createReids();

**byte**[] **serialize\_key** = SerializeUtil\_prjcli\_pkgcache.*serialize*(key.toString());

Jedis1.set(serialize\_key, SerializeUtil\_prjcli\_pkgcache.*serialize*(value));

Jedis1.expire(serialize\_key, 300); //def 5min

} **catch** (Exception **e**) {

*logger*.error(" mybatisredis cache.putObject ", e);

e.printStackTrace();

}

}

**public** **void** putObjectByThrdlocalCurKey(Object uMap) {

Object **curkey**= MybatisRedisCache.*curkey*.get();

String **string\_curkey** = curkey.toString().toLowerCase();

**if**(string\_curkey.contains("tmp") || string\_curkey.contains("temp") || string\_curkey.contains("count") || string\_curkey.contains("insert") || string\_curkey.contains("update") || string\_curkey.contains("delete") || string\_curkey.contains("drop") || string\_curkey.contains("exist"))

**return**;

**byte**[] **key2** = SerializeUtil\_prjcli\_pkgcache.*serialize*(string\_curkey);

Jedis **Jedis1**= createReids();

**if**(!Jedis1.exists(key2))

putObject(curkey,uMap);

## 默认namespace up del操作清空缓存，做了拦截不让他自动清空，不让缓存失效。使用时间到期清空模式

# 缓存内容与执行顺序不对 增加**synchronized** 锁定

## 可能没有走缓存，导致curkey还是上次的。每次put玩clr curkey 并put增加sync锁，保证走完

## 每次getxx执行前去除curkey thredlocal遍量

MybatisRedisCache.*curkey*.set(**null**);

**synchronized** (CliSendAdviceBo.**class**) {

MybatisRedisCache.*curkey*.set(**null**);

m = getBCK01A(map); // cache

//ati set cache

Object **key**= MybatisRedisCache.*curkey*.get();

**new** MybatisRedisCache().putObjectByThrdlocalCurKey\_AsynProxy("getBCK01A",m);

//end set cache finish

}

Map **m** ;

**synchronized** (CliSendAdviceBo.**class**) {

m = getBCK01A(map); // cache

//ati set cache

**new** MybatisRedisCache().putObjectByThrdlocalCurKey(m);

//end set cache finish

}

## 增加keytag 教研校验 这个额 keytag 是mybati配置文件里面的 nanespace.sqlid

**public** **void** putObjectByThrdlocalCurKey\_AsynProxy(**final** String key\_valideTag, **final** Object param\_shouldbelist) {

// if("a".equals("a"))

// return;

**final** Object **curkey1** = MybatisCacheWzGuava.*curkey*.get();

AsynUtil\_prjcli.*execMeth\_Async*(**new** FutureTask<>(**new** Callable() {

@Override

**public** Object call() **throws** Exception {

putObjectByThrdlocalCurKey(key\_valideTag, curkey1, param\_shouldbelist);

**return** **null**;

}

}), "threadName");

}

/\*\*

\*

\* **@param** param\_shouldbelist

\* **@param** m

\* **@param** parameter

\*/

**synchronized** **public** **void** putObjectByThrdlocalCurKey(String key\_valideTag, Object curkey1,

Object param\_shouldbelist) {

//// Object curkey = MybatisRedisCache.curkey.get();

**if** (curkey1 == **null**)

**return**;

**if** (curkey1.toString().contains("getBCK01A"))

System.***out***.println("dbg");

**if** (key\_valideTag.length() > 0)

**if** (!curkey1.toString().toLowerCase().contains(key\_valideTag.toLowerCase().trim()))

**return**;

/clinical/src/main/java/com/cnhis/cloudhealth/clinical/util/cache/MybatisCacheWzGuava.java